MC68PM302

Integrated
Multiprotocol Processor
with PCMCIA
Reference Manual

PREFACE

The complete documentation package for the MC68PM302 consists of the MC68PM302RM/AD, *MC68PM302 Low Power Integrated Multiprotocol Processor with PC-MCIA Interface Reference Manual*, M68000PM/AD, *MC68000 Family Programmer's Reference Manual,* MC68302UM/AD, *MC68302 Integrated Multiprotocol Processor User's Manual,* and the MC68PM302/D, *MC68PM302 Low Power Integrated Multiprotocol Processor with PCMCIA Interface Product Brief*.

The *MC68PM302 Low Power Integrated Multiprotocol Processor with PCMCIA Interface Reference Manual* describes the programming, capabilities, registers, and operation of the MC68PM302 that differ from the original MC68302; the *MC68000 Family Programmer's Reference Manual* provides instruction details for the MC68PM302; and the *MC68PM302 Low Power Integrated Multiprotocol Processor with PCMCIA Interface Product Brief* provides a brief description of the MC68PM302 capabilities.

The *MC68302 Integrated Multiprotocol Processor User's Manual User's Manual* is required, since the *MC68PM302 Low Power Integrated Multiprotocol Processor with PCMCIA Interface Reference Manual* only describes the new features of the MC68PM302.

This user's manual is organized as follows:

ELECTRONIC SUPPORT:

The Technical Support BBS, known as AESOP (Application Engineering Support Through On-Line Productivity), can be reach by modem or the internet. AESOP provides commonly asked application questons, latest device errata, device specs, software code, and many other useful support functions.

Modem: Call 1-800-843-3451 (outside US or Canada 512-891-3650) on a modem that runs at 14,400 bps or slower. Set your software to N/8/1/F emulating a vt100.

Internet: This access is provided by telneting to pirs.aus.sps.mot.com [129.38.233.1] or through the World Wide Web at http://pirs.aus.sps.mot.com.

— Sales Offices —

For questions or comments pertaining to technical information, questions, and applications, please contact one of the following sales offices nearest you.

UNITED STATES

ALABAMA, Huntsville	(205) 464-6800
ARIZONA, Tempe	(602) 897-5056
CALIFORNIA, Agoura Hills	(818) 706-1929
CALIFORNIA, Los Angeles	(310) 417-8848
CALIFORNIA, Irvine	(714) 753-7360
CALIFORNIA, Rosevllle	(916) 922-7152
CALIFORNIA, San Diego	(619) 541-2163
CALIFORNIA, Sunnyvale	(408) 749-0510
COLORADO, Colorado Springs	(719) 599-7497
COLORADO, Denver	(303) 337-3434
CONNECTICUT, Wallingford	(203) 949-4100
FLORIDA, Maitland	(407) 628-2636
FLORIDA, Pompano Beach/	
Fort Lauderdal	(305) 486-9776
FLORIDA, Clearwater	(813) 538-7750
GEORGIA, Atlanta	(404) 729-7100
IDAHO, Boise	(208) 323-9413
ILLINOIS, Chicago/Hoffman Estates	(708) 490-9500
INDIANA, Fort Wayne	(219) 436-5818
INDIANA, Indianapolis	(317) 571-0400
INDIANA, Kokomo	(317) 457-6634
IOWA, Cedar Rapids	(319) 373-1328
KANSAS, Kansas City/Mission	(913) 451-8555
MARYLAND, Columbia	(410) 381-1570

MASSACHUSETTS, Marborough	(508) 481-8100
MASSACHUSETTS, Woburn	(617) 932-9700
MICHIGAN, Detroit	(313) 347-6800
MINNESOTA, Minnetonka	(612) 932-1500
MISSOURI, St. Louis	(314) 275-7380
NEW JERSEY, Fairfield	(201) 808-2400
NEW YORK, Fairport	(716) 425-4000
NEW YORK, Hauppauge	(516) 361-7000
NEW YORK, Poughkeepsie/Fishkill	(914) 473-8102
NORTH CAROLINA, Raleigh	(919) 870-4355
OHIO, Cleveland	(216) 349-3100
OHIO, Columbus Worthington	(614) 431-8492
OHIO, Dayton	(513) 495-6800
OKLAHOMA, Tulsa	(800) 544-9496
OREGON, Portland	(503) 641-3681
PENNSYLVANIA, Colmar	(215) 997-1020
Philadelphia/Horsham	(215) 957-4100
TENNESSEE, Knoxville	(615) 690-5593
TEXAS, Austin	(512) 873-2000
TEXAS, Houston	(800) 343-2692
TEXAS, Plano	(214) 516-5100
VIRGINIA, Richmond	(804) 285-2100
WASHINGTON, Bellevue	(206) 454-4160
Seattle Access	(206) 622-9960
WISCONSIN, Milwaukee/Brookfield	(414) 792-0122

CANADA

BRITISH COLUMBIA, Vancouver	(604) 293-7605
ONTARIO, Toronto	(416) 497-8181
ONTARIO, Ottawa	(613) 226-3491
QUEBEC, Montreal	(514) 731-6881

INTERNATIONAL

AUSTRALIA, Melbourne	(61-3)887-0711
AUSTRALIA, Sydney	(61(2)906-3855
BRAZIL, Sao Paulo	55(11)815-4200
CHINA, Beijing	86 505-2180
FINLAND, Helsinki	358-0-35161191
Car Phone	358(49)211501
FRANCE, Paris/Vanves	33(1)40 955 900
GERMANY, Langenhagen/ Hanover	49(511)789911
GERMANY, Munich	49 89 92103-0
GERMANY, Nuremberg	49 911 64-3044
GERMANY, Sindelfingen	49 7031 69 910
GERMANY,Wiesbaden	49 611 761921
HONG KONG, Kwai Fong	852-4808333
Tai Po	852-6668333
INDIA, Bangalore	(91-812)627094
ISRAEL, Tel Aviv	972(3)753-8222
ITALY, Milan	39(2)82201
JAPAN, Aizu	81(241)272231
JAPAN, Atsugi	81(0462)23-0761
JAPAN, Kumagaya	81(0485)26-2600
JAPAN, Kyushu	81(092)771-4212
JAPAN, Mito	81(0292)26-2340
JAPAN, Nagoya	81(052)232-1621
JAPAN, Osaka	81(06)305-1801
JAPAN, Sendai	81(22)268-4333
JAPAN, Tachikawa	81(0425)23-6700
JAPAN, Tokyo	81(03)3440-3311
JAPAN, Yokohama	81(045)472-2751
KOREA, Pusan	82(51)4635-035
KOREA, Seoul	82(2)554-5188
MALAYSIA, Penang	60(4)374514
MEXICO, Mexico City	52(5)282-2864
MEXICO, Guadalajara	52(36)21-8977
Marketing	52(36)21-9023
Customer Service	52(36)669-9160
NETHERLANDS, Best	(31)49988 612 11
PUERTO RICO, San Juan	(809)793-2170
SINGAPORE	(65)2945438
SPAIN, Madrid	34(1)457-8204
or	34(1)457-8254
SWEDEN, Solna	46(8)734-8800
SWITZERLAND, Geneva	41(22)7991111
SWITZERLAND, Zurich	41(1)730 4074
TAIWAN, Taipei	886(2)717-7089
THAILAND, Bangkok	(66-2)254-4910
UNITED KINGDOM, Aylesbury	44(296)395-252

FULL LINE REPRESENTATIVES

COLORADO, Grand Junction	
Cheryl Lee Whltely	(303) 243-9658
KANSAS, Wichita	
Melinda Shores/Kelly Greiving	(316) 838 0190
NEVADA, Reno	
Galena Technology Group	(702) 746 0642
NEW MEXICO, Albuquerque	
S&S Technologies, Inc.	(505) 298-7177
UTAH, Salt Lake City	
Utah Component Sales, Inc.	(801) 561-5099
WASHINGTON, Spokane	
Doug Kenley	(509) 924-2322
ARGENTINA, Buenos Aires	
Argonics, S.A.	(541) 343-1787

HYBRID COMPONENTS RESELLERS

Elmo Semiconductor	(818) 768-7400
Minco Technology Labs Inc.	(512) 834-2022
Semi Dice Inc.	(310) 594-4631

TABLE OF CONTENTS

Paragraph Number | Title | Page Number

Section 1
Introduction

Section 2
Configuration, Clocking, Low Power Modes, and Internal Memory Map

Section 3
System Integration Block (SIB)

Section 4
Communications Processor (CP)

Section 5
PCMCIA Controller

Section 6
Signal Description

Section 7

Paragraph Number | Title | Page Number

Electrical Characteristics

Section 8
Mechanical Data and Ordering Information

SECTION 1
INTRODUCTION

Motorola has developed a derivative version of the well-known MC68302 integrated multi-protocol processor (IMP) called the MC68PM302. The PM302 has two modes of operation. In the first mode, it functions as an enhanced MC68302 with a new static 68000 core, a new timer and low power modes, and additional parallel I/O pins. In the second mode, it functions as the same enhanced MC68302, but with PCMCIA and 16550 UART functionality instead of the additional parallel I/O pins. The PM302 is packaged in a low profile 144 TQFP that is suitable for use in Type II PCMCIA cards.

The Personal Computer Memory Card International Association (PCMCIA) has implemented a set of specifications for small form-factor cards. These cards were previously specified by a version number of the spec to which they complied. After the specification 2.1, these cards were named “PC Cards.” The PCMCIA functionality on the PM302 is compliant with the specification approved in October 1994.

In addition, the PM302 contains a parallel interface and register set identical to the 16550 UART used in IBM-compatible PCs. This functionality is available when the PCMCIA mode is enabled, and shares pins with the PCMCIA interface.

The document fully describes all the differences between the PM302 and the regular 68302. Any feature not described in this document will operate as described in the *MC68302 User’s Manual*. In addition this document contains the full set of electrical descriptions for the PM302.

1.1 BLOCK DIAGRAM

The block diagram is shown in Figure 1-1.

1.2 FEATURES

The features of the PM302 are as follows. The items in **bold face** type show major differences from the MC68302, although a complete list of differences is given in section 1.4 on page 4.

- On-Chip **Static 68000 Core** Supporting a 16- or 8-Bit M68000 Family-System
- SIB Including:

  Independent Direct Memory Access (IDMA) Controller.

  Interrupt Controller with Two Modes of Operation

  Parallel Input/Output (I/O) Ports, some with Interrupt Capability

  On-Chip 1152-Byte Dual-Port RAM

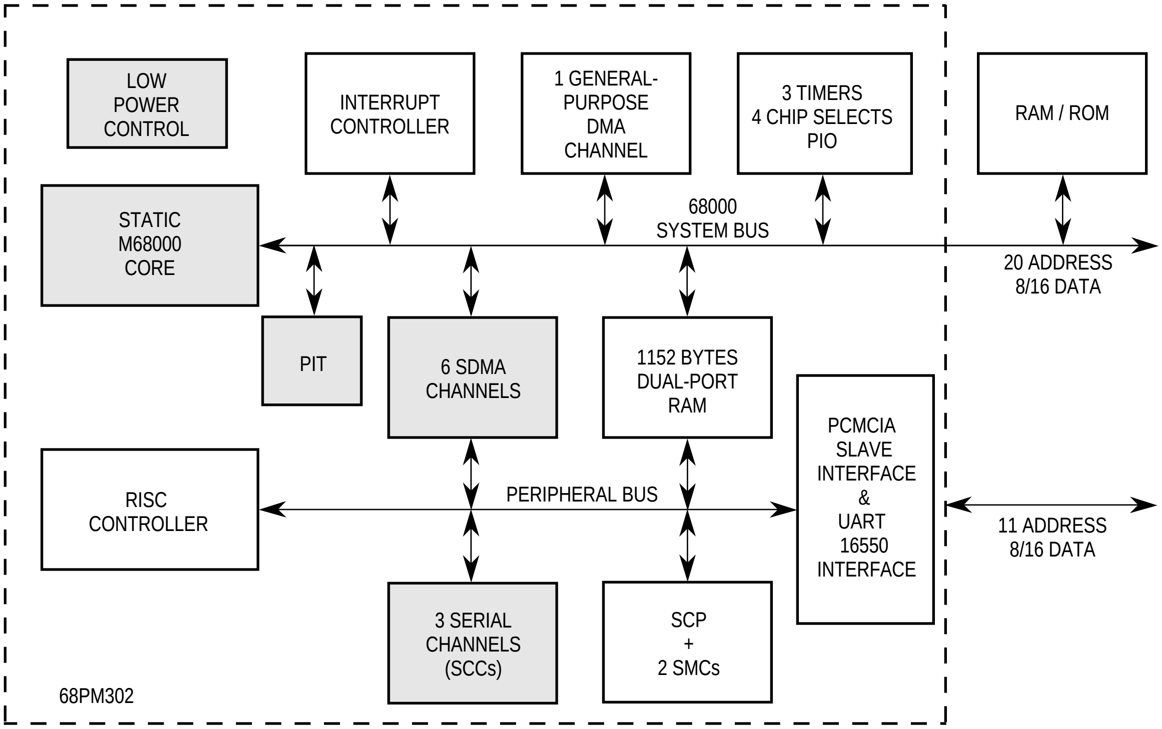


Figure 1-1. MC68PM302 Block Diagram

Three Timers Including a Watchdog Timer

New Periodic Interrupt Timer (PIT)

Four Programmable Chip-Select Lines with Wait-State Generator Logic

Programmable Address Mapping of the Dual-Port RAM and IMP Registers

On-Chip Clock Generator with Output Signal

Glueless Interface to EPROM, SRAM, Flash EPROM, and EEPROM

Allows Boot in 8-Bit Mode, and Running Switch to 16-bit Mode

Allows System Clock to be Generated from 32 kHz or 4 MHz Crystals in Addition to a Full Speed Clock Oscillator

System Control:
System Status and Control Logic
Disable CPU Logic (Slave Mode Operation)
Hardware Watchdog
New Low-Power (Standby) Modes in μA Range With Wake-up From Two Pins or PIT Timer
Freeze Control for Debugging
DRAM Refresh Controller

- CP Including:

Main Controller (RISC Processor)

Three Independent Full-Duplex Serial Communications Controllers (SCCs)

Supporting Various Protocols:

High-Level/Synchronous Data Link Control (HDLC/SDLC)
Universal Asynchronous Receiver Transmitter (UART)
Binary Synchronous Communication (BISYNC)
Transparent Modes
Autobaud Support Instead of DDCMP and V.110
Boot from SCC Capability

Six Serial DMA Channels for the Three SCCs

Flexible Physical Interface Accessible by SCCs Including ISDN Support:

Motorola Interchip Digital Link (IDL)
General Circuit Interface (GCI, Also Known as IOM[1]-2)
Pulse Code Modulation (PCM) Highway Interface
Nonmultiplexed Serial Interface (NMSI) Implementing Standard Modem Signals

SCP for Synchronous Communication

Two Serial Management Controllers (SMCs) To Support IDL and GCI Auxiliary Channels

- **PCMCIA Controller**

Supports Slave Interface Having 8 or 16-bit Data Pins

Compatible with the PC Card Classic Specification Approved in October 1994

Support for Ring Detect and Other Indications

Support for Power Down Modes and Automatic Wakeup on Host Access

Card Information Structure (CIS) Size and Location In Memory Programmable

Supports DMA Accesses as Defined by PCMCIA

Supports Direct Access to Common Memory Space for High Speed Buffer Transfers

Security Option to Prevent Host Accesses to Sensitive Address Ranges

- **16550 Emulation Block**

Complete H/W and S/W Emulation of the 16550 UART

DMA Supported for Transfer of 16550 Data Over 68K bus

Speed of Data Transfer Can Be Matched to Traditional 16550 UART

High Speed Data Transfer up to 2 Mbps Possible

- Initial Offering Features 16.67 and 20 MHz Versions at 5V and 3.3V
- 144 Pin Thin Quad Flat Pack (TQFP) Packaging

1.3 PM302 APPLICATIONS

The PM302 excels in several applications areas.

[1] IOM is a trademark of Siemens AG

First, the PM302 excels in low power and portable applications. The inclusion of a static 68000 core coupled with the low power modes built into the device make it ideal for handheld, or other low power applications. The new 32 kHz or 4 MHz PLL option greatly reduces the total power budget of the designer's board, and allows the PM302 to be an effective device in low power systems. The PM302 can then optionally generate a full frequency clock for use by the rest of the board. During low power modes, the new periodic interrupt timer (PIT) allows the device to be woken up at regular intervals. In addition, two pins allow the device to be woken up from low power modes. These features make the device useful as an upgrade to the existing 68302.

Second, the performance, cost, and low power usage of the PM302 is an ideal processor to be used on PCMCIA cards. It will perform the controller functions of a V.34 fax/data modem in conjunction with a separate data pump. With its PCMCIA and ISDN functionality on-chip, it can be gluelessly connected to an S/T or U interface device to perform the functions of a ISDN terminal adaptor. It can be used as a MAC controller for IEEE 802.11 wireless LANs. It may also be used as a wireless WAN PCMCIA controller for CDPD, Ardis, Mobitex, or digital cellular data applications.

1.4 PM302 DIFFERENCES FROM OTHER DEVICES

The PM302 can be considered as a superset of another derivative called the MC68LC302. The LC302 does not have the PCMCIA and 16550 features, and only contains 2 SCCs. The PM302 can also be considered to be a subset of the MC68356. The 356 adds a full 56002 DSP with memory and, in one of its versions, contains a full V.34 fax/data data pump running on the DSP.

When the PM302 is operating in its PCMCIA mode, certain functions existing on the original 68302 are no longer available. The major differences are listed below.

- External masters are not able to take the bus away from the PM302 except through a simple scheme using the $\overline{\text{HALT}}$ pin. This restriction does not apply to using the PM302 is in CPU disabled mode (slave mode), in which case $\overline{\text{BR}}$, $\overline{\text{BG}}$, and $\overline{\text{BGACK}}$ are all available.
- Although the Independent DMA (IDMA) is still available, the external IDMA request pins ($\overline{\text{DACK}}$, and $\overline{\text{DONE}}$) have been eliminated.
- The DDCMP and V.110 protocols have been removed.
- Four address lines have been eliminated, giving a total of 20 address lines. However, the PM302 supports more than a 1 MB addressing range, since each of the four chip selects still decodes a 24-bit address. This allows a total of 4 MB to be addressed.
- The $\overline{\text{UDS}}$, $\overline{\text{LDS}}$, and R/$\overline{\text{W}}$ pins are not available except in slave mode, where they replace the $\overline{\text{WEH}}$, $\overline{\text{WEL}}$, and $\overline{\text{OE}}$ pins. Instead, the new pins $\overline{\text{WEH}}$, $\overline{\text{WEL}}$, and $\overline{\text{OE}}$ have been defined for glueless interfacing to memory.
- PA12 is now muxed with the MODCLK pin, which is associated with the 32 kHz or 4 MHz PLL. The MODCLK pin is sampled at reset, and then becomes the PA12 pin.
- $\overline{\text{BCLR}}$, $\overline{\text{RMC}}$ and $\overline{\text{BRG1}}$ pins have been eliminated
- New VCCsyn, GNDsyn, and XFC pins have been added in support of the on-chip PLL.

- For purposes of emulation support only, a special 180 PGA version is supported. This version adds back the FC2-0, $\overline{\text{IAC}}$, $\overline{\text{FRZ}}$, and $\overline{\text{AVEC}}$ pins. The FC2-0 pins allow bus cycles to be distinguished between program and data accesses, interrupt cycles, etc. The $\overline{\text{IAC}}$, $\overline{\text{FRZ}}$, and $\overline{\text{AVEC}}$ pins are provided so that emulation vendors can quickly retrofit their existing 68302 emulator designs to support the PM302.

When the PM302 is not used in its PCMCIA mode, most of the original 68302 pins are returned to the device. However, even though the 68302 is also offered in a 144 TQFP, it was not possible to make the PM302 exactly pin compatible with the original 68302. Thus, the PM302 should not be used as a drop-in replacement for the 68302 in a 144 TQFP.

SECTION 2
CONFIGURATION, CLOCKING, LOW POWER MODES, AND INTERNAL MEMORY MAP

The MC68PM302 integrates a high-speed M68000 processor with multiple communications peripherals. The provision of direct memory access (DMA) control and link layer management with the serial ports allows high throughput of data for communications-intensive applications, such as basic rate Integrated Services Digital Network (ISDN).

The MC68PM302 can operate either in the full MC68000 mode with a 16-bit data bus or in the MC68008 mode with an 8-bit data bus by connecting the BUSW/PC_ABUF pin low during reset.

NOTE

> The BUSW pin is static and is not intended to be used for dynamic bus sizing. Instead the BSW and BSWEN bits in the PPR register should be used to switch the bus width after reset.

Refer to the MC68000UM/AD, *M68000 8-/16-/32-Bit Microprocessors User's Manual*, and the MC68302UM/AD, *MC68302 Integrated Multiprotocol Processor User's Manual*, for complete details of the on-chip microprocessor including the programming model and instruction set summary. Throughout this manual, references may use the notation M68000, meaning all devices belonging to this family of microprocessors, or the notation MC68000, MC68008, meaning the specific microprocessor products.

This section is intended to describe configuration of the MC68PM302 and the differences between thePM302 and the MC68000 and the MC68302.This section also includes tables that show the registers of the IMP portion of the MC68PM302. All of the registers are memory mapped into the 68000 space

2.1 MC68PM302 AND MC68302 SIGNAL DIFFERENCES

When the PM302 is operating in its PCMCIA mode, certain functions existing on the original 68302 are no longer available. The major differences are listed below.

- External masters are not able to take the bus away from the PM302 except through a simple scheme using the $\overline{\text{HALT}}$ pin. This restriction does not apply to using the PM302 is in CPU disabled mode (slave mode), in which case $\overline{\text{BR}}$, $\overline{\text{BG}}$, and $\overline{\text{BGACK}}$ are all available.
- Although the Independent DMA (IDMA) is still available, the external IDMA pins ($\overline{\text{DACK}}$ and $\overline{\text{DONE}}$) have been eliminated.

- Four address lines have been eliminated, giving a total of 20 address lines. However, the PM302 supports more than a 1 MB addressing range, since each of the four chip selects still decodes a 24-bit address. This allows a total of 4 MB to be addressed.
- The FC2-0 pins have been eliminated. In CPU disabled mode (slave mode), these signals are internally driven to a value of 5 (Supervisor Data function code).
- The $\overline{\text{UDS}}$, $\overline{\text{LDS}}$, and R/$\overline{\text{W}}$ pins are not available except in slave mode, where they replace the WEH, WEL, and OE pins. The new pins $\overline{\text{WEH}}$, $\overline{\text{WEL}}$, and $\overline{\text{OE}}$ are available in enable CPU mode. Their functionalities have been defined for glueless interfacing to memory.
- PA12 is now muxed with the MODCLK pin, which is associated with the 32 kHz or 4 MHz PLL. The MODCLK pin is sampled at reset, and then becomes the PA12 pin.
- New VCCsyn, GNDsyn, and XFC pins have been added in support of the on-chip PLL.
- For purposes of emulation support only, a special 180 PGA version is supported. This version adds back the FC2-0, $\overline{\text{IAC}}$, $\overline{\text{FRZ}}$, and $\overline{\text{AVEC}}$ pins.
- PA7 is sampled during reset to enable Boot from SCC.

When the PM302 is not used in its PCMCIA mode, most of the original 68302 pins are returned to the device. However, even though the 68302 is also offered in a 144 TQFP, it was not possible to make the PM302 exactly pin compatible with the original 68302. Thus, the PM302 should not be used as a drop-in replacement for the 68302 in a 144 TQFP.

2.2 IMP CONFIGURATION CONTROL

A number of reserved entries in the external M68000 exception vector table are used as addresses for internal system configuration registers. See Table 2-1, System Configuration Registers.

The BAR entry contains the BAR described in this section. The SCR entry contains the SCR described in Section 3 System Integration Block (SIB).

Figure 2-1 shows all the IMP on-chip addressable locations and how they are mapped into system memory.

The on-chip peripherals, including those peripherals in both the communications processor (CP) and system integration block (SIB), require a 4K-byte block of address space. This 4K-byte block location is determined by writing the intended base address to the BAR in supervisor data space (FC = 5). In slave mode the FC2-0 pins are internally driven by the MC68PM302 to supervisor data space. In enable CPU mode, the Status Register of the M68000 should be set to supervisor mode, in order to access the supervisor data space.

After a total system reset, the on-chip peripheral base address is undefined, and it is not possible to access the on-chip peripherals at any address until BAR is written. The BAR and the SCR can always be accessed at their fixed addresses.

NOTE

> The BAR and SCR registers are internally reset only when a total system reset occurs by the simultaneous assertion of RESET

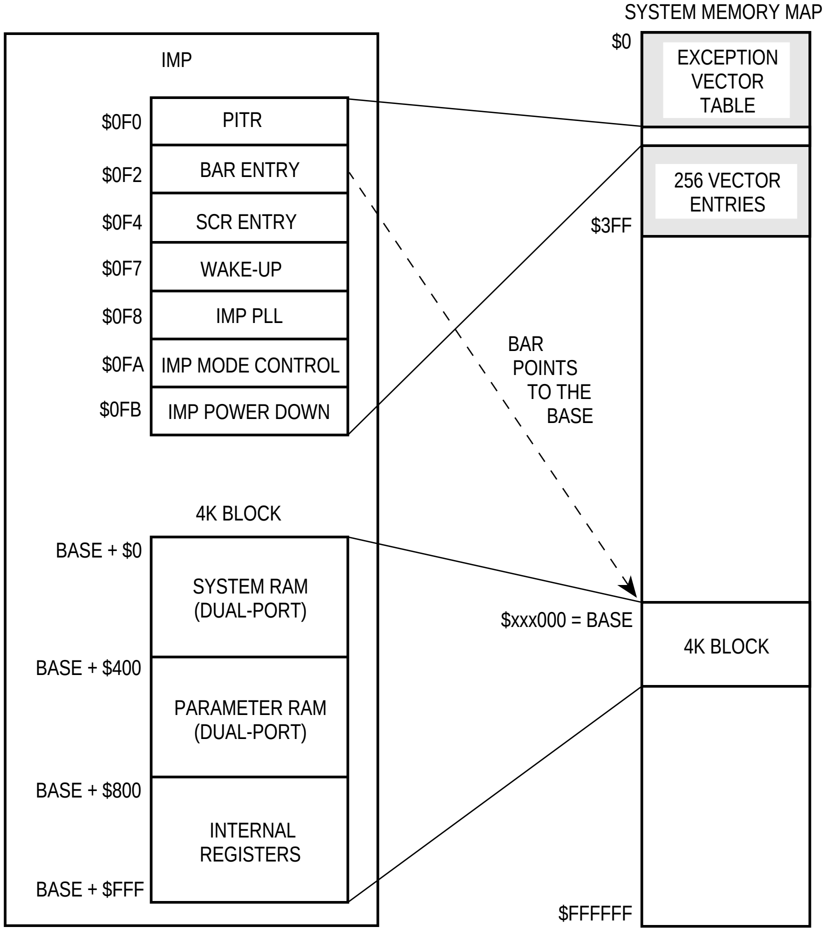


Figure 2-1. IMP Configuration Control

and HALT. The chip-select (CS) lines are not asserted on accesses to these locations. Thus, it is very helpful to use CS lines to select external ROM/RAM that overlaps the BAR and SCR register locations, since this prevents potential bus contention.

NOTE

In 8-bit system bus operation, IMP accesses are not possible until the low byte of the BAR is written. Since the MOVE.W instruction writes the high byte followed by the low byte, this instruction guarantees the entire word is written.

Do not assign other devices on the system bus an address that falls within the address range of the peripherals defined by the BAR. If this happens, a $\overline{\text{BERR}}$ is generated to the core (if the address decode conflict enable (ADCE) bit is set) and the address decode conflict (ADC) bit in the SCR is set.

2.2.1 Base Address Register (BAR)

The BAR is a 16-bit, memory-mapped, read-write register consisting of the high address bits, the compare function code bit, and the function code bits. Upon a total system reset, its value may be read as $BFFF, but its value is not valid until written by the user. The address of this register is fixed at $0F2 in supervisor data space. BAR cannot be accessed in user data space.

15 13	12	11 0
FC2–FC0	CFC	BASE ADDRESS 23 22 21 20 19 18 17 16 15 14 13 12

Bits 15–13—FC2–FC0

The FC2–FC0 field is contained in bits 15–13 of the BAR. These bits are used to set the address space of 4K-byte block of on-chip peripherals. The address compare logic uses these bits, dependent upon the CFC bit, to cause an address match within its address space. When the core is enabled, the function code bits will be driven by the core to indicate the type of cycle in process. In disable CPU mode, the FC pins are not present unless the PCMCIA is disabled (they are internally driven to 5). In this situation, the user does not have any control over how the FC signals are driven, and it is recommended that the user write these bits to zero and write the CFC bit to zero to disable the FC comparison.

NOTE

Do not assign this field to the M68000 core interrupt acknowledge space (FC2–FC0 = 7).

CFC—Compare Function Code

0 = The FC bits in the BAR are ignored. Accesses to the IMP 4K-byte block occur without comparing the FC bits.

1 = The FC bits in the BAR are compared. The address space compare logic uses the FC bits to detect address matches.

Bits 11–0—Base Address

The high address field is contained in bit 11–0 of the BAR. These bits are used to set the starting address of the dual-port RAM. The address compare logic uses only the most significant bits to cause an address match within its block size. Even though A23-20 are signals are not available, they are driven internally by the core, or driven to zeroes in disable CPU mode or when $\overline{\text{HALT}}$ has been asserted by an external master.

2.3 SYSTEM CONFIGURATION REGISTERS

Four entries in the M68000 exception vectors table (located in low RAM) are reserved for the addresses of system configuration registers (see Table 2-1). These registers have seven addresses within $0F0-$0FF. The MC68PM302 uses one of the IMP 32-bit reserved spaces for 3 registers added for the MC68PM302. These registers are used to control the

PLL, clock generation and low power modes. See 2.4 Clock Generation and Low Power Control.

Table 2-1. System Configuration Registers

Address	Name	Width	Description	Reset Value
$0F0	PITR	16	Periodic Interrupt Timer Register	0000
$0F2	BAR	16	Base Address Register	BFFF
$0F4	SCR	24	System Control Register	0000 0F
$0F7	IWUCR	8	IMP Wake-Up Control Register	00
$0F8	IPLCR	16	IMP PLL Control Register	
$0FA	IOMCR	8	IMP Operations Mode Control Register	00
$0FB	IPDR	8	IMP Power Down Register	00
$0FC	RES	32	Reserved	

2.4 CLOCK GENERATION AND LOW POWER CONTROL

The MC68PM302 includes a clock circuit that consists of crystal oscillator drive circuit capable of driving either an external crystal or accepting an oscillator clock, a PLL clock synthesizer capable of multiplying a low frequency clock or crystal such as a 32-kHz watch crystal up to the maximum clock rate of each processor, and a low power divider which allows dynamic gear down and gear up of the system clock for each processor on the fly.

- **On-Chip Clock Synthesizers (with output system clocks)**

  —Oscillator Drive Circuits and Pins

  —PLL Clock Synthesizer Circuits with Low Power Output Clock Divider Block.
- **Low Power Control Of IMP**

  —Slow-Go Modes using PLL Clock Divider Blocks

  —Varied Low Power STOP Modes for Optimizing Wake-Up Time to Low Power Mode Power Consumption: Stand-By, Doze and STOP.

2.4.1 PLL and Oscillator Changes to IMP

The oscillator that was on the MC68302 has been replaced by the new clock synthesizer described in this section. The registers related to the oscillator have been either removed or changed according to the description below. Several control bits are still available but have new locations.

The low power modes on the MC68302 have changed completely and will be discussed later in 2.4.4.1 IMP Low Power Modes

2.4.1.1 CLOCK CONTROL REGISTER. The clock control register address $FA is not implemented on the MC68PM302. This register location has been reassigned to the IOMCR and ICKCR registers. The clock control register bits have been reassigned as follows:

CLKO Drive Options (CLKOMOD1–2)

These bits are now in the IMP clock control register (IPLCR) on the MC68PM302, see 2.4.3.4.2 IMP PLL and Clock Control Register (IPLCR).

Three-State TCLK1 (TSTCLK1)

This bit is now in the BRGP register on the MC68PM302, see section 4.3.2 Baud Rate Generator Pins Register–BRGP.

Three-State RCLK1 (TSRCLK1)

This bit is now in the BRGP register on the MC68PM302,see section 4.3.2 Baud Rate Generator Pins Register–BRGP.

Disable BRG1 (DISBRG1)

This bit has been removed since the BRG1 pin was removed.

2.4.2 MC68PM302 System Clock Generation

Figure 2-3, the MC68PM302 system clock schematic, shows the IMP clock synthesizer. The block includes an on-chip oscillator, a clock synthesizer, and a low-power divider, which allows a comprehensive set of options for generating the system clock. The choices offer many opportunities to save power and system cost, without sacrificing flexibility and control. In addition to performing frequency multiplication, the PLL block can also provide EXTAL to CLKO skew elimination, and dynamic low power divides of the output PLL system clock.

Clock source and default settings are determined during the reset of the IMP. The MC68PM302 decodes the MODCLK and VCCSYN pins and the value of these pins determines the initial clocking for the part. Further changes to the clocking scheme can be made by software. After reset, the 68000 core can control the IMP clocking through the following registers:

1. IMP Operation Mode Control Register, IOMCR (2.4.4.1.6 IMP Operation Mode Control Register (IOMCR)).
2. IMP PLL and Clock Control Register, IPLCR (2.4.3.4 Frequency Multiplication).
3. IMP Interrupt Wake-Up Control Register, IWUCR (2.4.4.2.4 IMP Wake-Up Control Register – IWUCR).
4. Periodic Interrupt Timer Register, PITR (See Section 3 System Integration Block (SIB)).

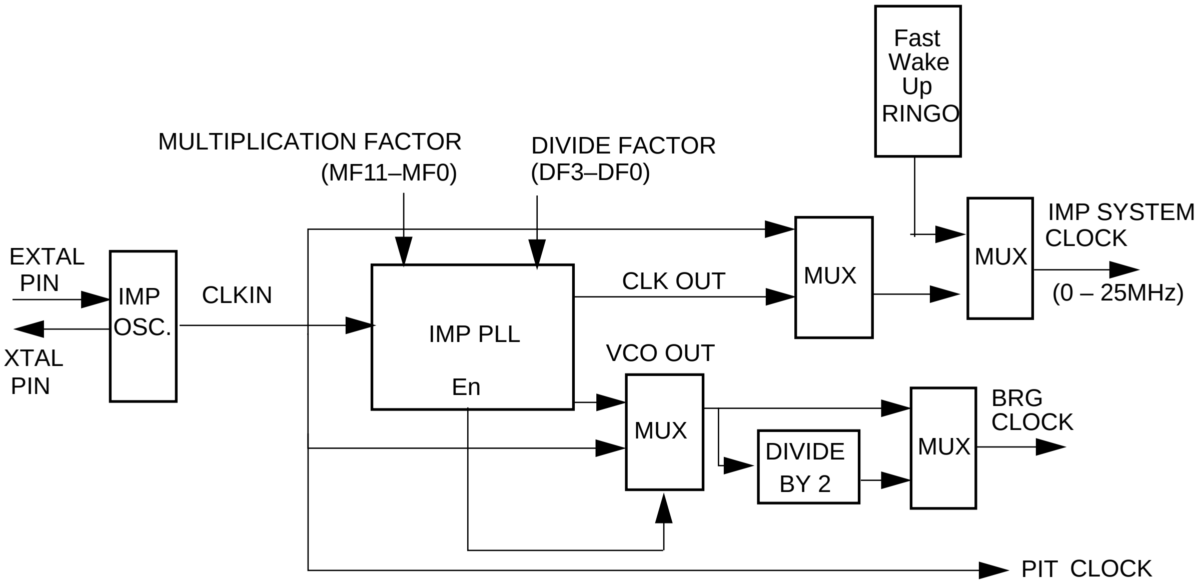


Figure 2-2. MC68PM302 PLL Clock Generation Schematic

2.4.2.1 DEFAULT SYSTEM CLOCK GENERATION. During the assertion of hardware reset, the value of the MODCLK and VCCSYN input pins determine the initial PLL settings according to Table 2-2. After the deassertion of reset, these pins are ignored.

The MODCLK and VCCSYN pins control the IMP clock selection at hardware reset. The IMP PLL can be enabled or disabled at reset only and the multiplication factor preset to support different industry standard crystals. After reset, the multiplication factor can be changed in the IPLCR register, and the IMP PLL divide factor can be set in the IOMCR register.

NOTE

The IMP input frequency ranges are limited to between 25 kHz and the maximum operating frequency, and the PLL output frequency range before the low power divider is limited to between 10 MHz and the maximum system clock frequency (25 MHz).

Table 2-2. Default System Clock Generation

CSelect	VCCSYN MODCLK	Example IMP EXTAL Freq.	IMP PLL	IMP MF+1	IMP System Clock
0	0X	25 MHz	Disabled	x	IMP EXTAL
0	10	4.192 MHz	Enabled	4	IMP EXTALx4
0	11	32.768 kHz	Enabled	401	IMP EXTALx401

Note:
By loading the IPLCR register the user can change the multiplication factor of the PLL after RESET.
By loading the IOMCR register, the user can change the power saving divide factor of the IMP PLL.

NOTE

It is not possible to start the system with PLL disabled and then enable the PLL with software programming.

2.4.3 IMP System Clock Generation

2.4.3.1 SYSTEM CLOCK CONFIGURATION. The IMP has an on-chip oscillator and phased locked loop (Figure 2-2). These features provide flexible ways to save power and reduce system cost. The operation of the clock generation circuitry is determined by the following registers.

The IMP Operation Mode Control Register, IOMCR in 2.4.4.1.6 IMP Operation Mode Control Register (IOMCR).

The IMP PLL and Clock Control Register, IPLCR in 2.4.3.3 Phase-Locked Loop (PLL).

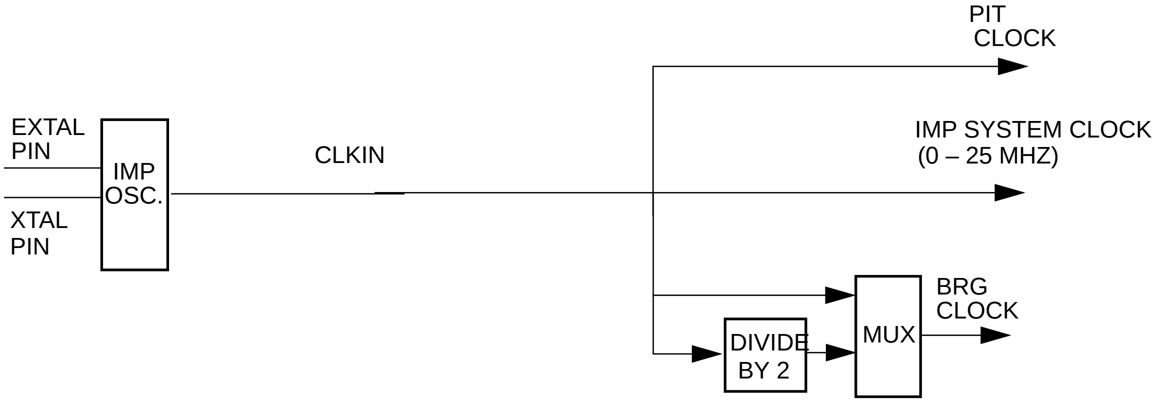


Figure 2-3. IMP System Clocks Schematic - PLL disabled

Figure 2-2 shows the IMP system clocks schematic with the IMP PLL enabled. Figure 2-3 shows the IMP system clocks schematic with the IMP PLL disabled.

The clock generation features of the IMP are discussed in the following paragraphs.

2.4.3.2 ON-CHIP OSCILLATOR. A 32.768-kHz watch crystal provides an inexpensive reference, but the EXTAL reference crystal frequency can be any frequency from 25 kHz to 6.0 MHz. Additionally, the system clock frequency can be driven directly into the EXTAL pin. In this case, the EXTAL frequency should be the exact system frequency desired (0 to Maximum operating frequency), and the XTAL pin should be left floating. Figure 2-4 shows all the external connections required for the on-chip oscillator (as well as the PLL VCC and GND connections).

2.4.3.3 PHASE-LOCKED LOOP (PLL). The IMP PLL's main function is frequency multiplication. The phase-locked loop takes the CLKIN frequency and outputs a high-frequency source used to derive the general system frequency of the IMP. The IMP PLL is comprised of a phase detector, loop filter, voltage-controlled oscillator (VCO), and multiplication block.

2.4.3.4 FREQUENCY MULTIPLICATION. The IMP PLL can multiply the CLKIN input frequency by any integer between 1 and 4096. The multiplication factor may be changed to the desired value by writing the MF11–MF0 bits in the IPLCR. When the IMP PLL multiplier is modified in software, the IMP PLL will lose lock, and the clocking of the IMP will stop until

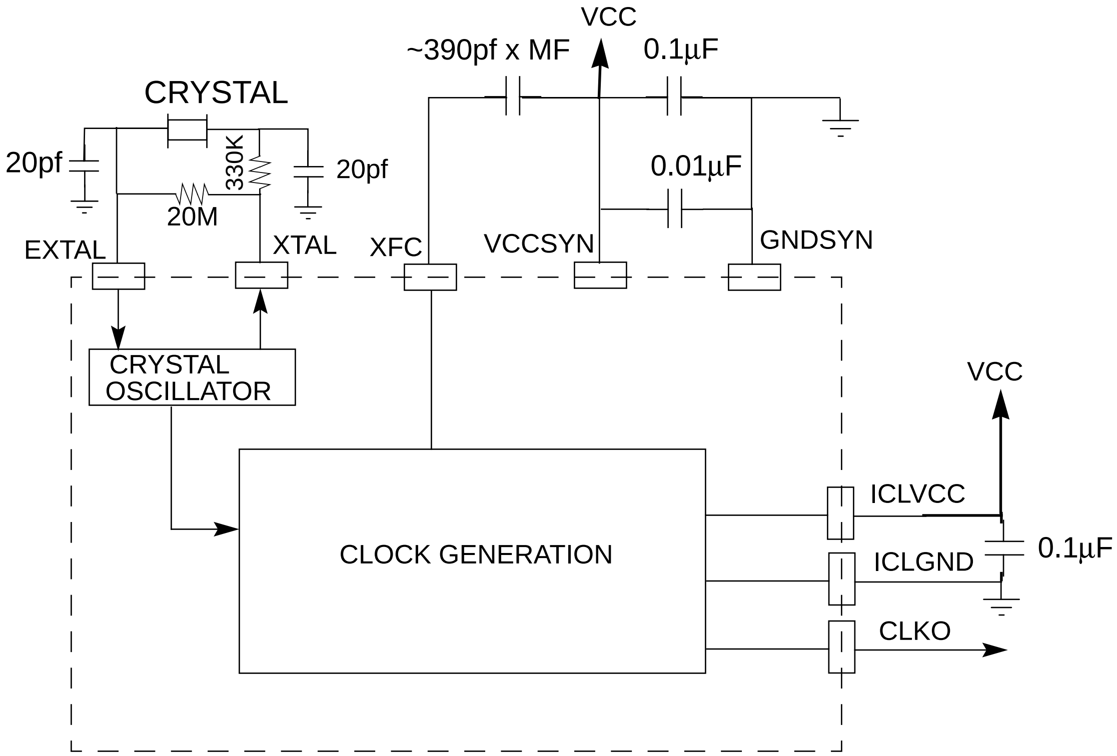


Figure 2-4. PLL External Components

lock is regained (worst case is 2500 EXTAL clocks). If an alteration in the system clock rate is desired without losing IMP PLL lock, the value in the low-power clock divider can be to modified to lower the system clock rate dynamically. The low power clock divider bits are located in the IOMCR register.

NOTE

If IMP PLL is enabled, the multiplication value must be large enough to result in the VCO clock being greater than 10 MHz.

2.4.3.4.1 Low Power PLL Clock Divider. The output of the IMP VCO is sent to a low power divider block. The clock divider can divide the output frequency of the VCO before it generates the system clock. The clock for the baud rate generators (BRGs) bypasses this clock divider.

The purpose of the clock divider is to allow the user to reduce and restore the operating frequency of the IMP without losing the IMP's PLL lock. Using the clock divider, the user can still obtain full IMP operation, but at a slower frequency. The BRG is not affected by the low power divider circuitry so previous BRG divider settings will not have to be changed when the divide factors are changed.

When the PLL low power divider bits (DF0–3) are programmed to a non-zero value, the IMP is in SLOW_GO mode. The selection and speed of the SLOW_GO mode may be changed at any time, with changes occurring immediately.

NOTE

> The IMP low power clock divider is active only if the IMP PLL is active.

The low-power divider block is controlled in the IOMCR. The default state of the low-power divider is to divide all clocks by 1.

If the low-power divider block is not used and the user is concerned that errant software could accidentally write the IOMCR, the user may set a write protection bit in IOMCR to prevent further writes to the register.

2.4.3.4.2 IMP PLL and Clock Control Register (IPLCR). IPLCR is a 16-bit read/write register used to control the IMP's PLL, multiplication factor and CLKO drive strength. This register is mapped in the 68000 bus space at address $0F8. If the 68000 bus is set to 8 bits (BUSW grounded at reset), during 8-bit accesses, changes to the IPLCR will take effect in the IMP PLL after loading the high byte of IPLCR (the low byte is written first). The WP bit in IPLCR is used as a protect mechanism to prevent erroneous writing. When this bit is set further accesses to the IPLCR will be blocked.

IMP PLL and Clock Control Register (IPLCR) $0F8

15	14	13	12	11	10	9	8
IPLWP	CLKOMOD0–1		PEN	MF11	MF10	MF9	MF8
RESET 0	0	0	VCCSYN	0	0	0	VCCSYN/ MODCLK

7	6	5	4	3	2	1	0
MF7	MF6	MF5	MF4	MF3	MF2	MF1	MF0
RESET VCCSYN/ MODCLK1	0	0	VCCSYN/ MODCLK	0	0	MODCLK	MODCLK

Read/Write

MF 11–0—Multiplication Factor

These bits define the multiplication factor that will be applied to the IMP PLL input frequency. The multiplication factor can be any integer from 1 to 4096. The system frequency is ((MF bits + 1) x EXTAL). The multiplication factor must be chosen to ensure that the resulting VCO output frequency will be in the range from 10 MHz to the maximum allowed clock input frequency (e.g. 20 MHz for a 20 MHz IMP).

The value 000 results in a multiplier value of 1. The value $FFF results in a multiplier value of 4096.

Any time a new value is written into the MF11–MF0 bits, the IMP PLL will lose the lock condition, and after a delay of 2500 EXTAL clocks, will relock. When the IMP PLL loses its lock condition, all the clocks that are generated by the IMP PLL are disabled. After hardware reset, the MF11–MF0 bits default to either 0, 3 or 400 ($190 hex) depending on the MODCLK and VCCSYN pins (giving a multiplication factor of 1, 4 or 401). If the multiplication factor is 401, then a standard 32.768 kHz crystal generates an initial general system clock of 13.14 MHz. If the multiplication factor is 4, then a standard 4.192 MHz crystal generates an initial general system clock of 16.768 MHz. The user would then write the MF bits or adjust the output frequency to the desired frequency.

NOTE

> Since the clock source for the periodic interrupt timer is CLKIN (see Figure 2-2), the PIT timer is not disturbed when the IMP PLL is in the process of acquiring lock.

PEN—PLL Enable Bit

The PEN bit indicates whether the IMP PLL is operating. This bit is written by the MC68PM302 based on the value of VCCSYN during reset. When the IMP PLL is disabled, the VCO is not operating in order to minimize power consumption. During hardware reset this bit is set if the VCCSYN pin specifies that the IMP PLL is enabled. The only way to clear PEN is to hold the VCCSYN pin low during a hardware reset.

0 = The IMP PLL is disabled. Clocks are derived directly from the EXTAL pin.
1 = The IMP PLL is enabled. Clocks are derived from the CLKOUT output of the PLL.

CLKODM0–1—CLKO Drive Mode 0–1

These bits control the output buffer strength of the CLKO pin. Those bits can be dynamically changed without generating spikes on the CLKO pin. Disabling CLKO will save power and reduce noise.

00 = Clock Out Enabled, Full-Strength Output Buffer.
01 = Clock Out Enabled, 2/3-Strength Output Buffer
10 = Clock Out Enabled, 1/3-Strength Output Buffer
11 = Clock Out Disabled (CLKO is driven high by internal pullup)

NOTE

> These IMP bits are in a different address location than in the MC68302, where they are located at address $FA (bits 15, 14).

IPLWP—IMP PLL Control Write Protect Bit

This bit prevents accidental writing into the IPLCR. After reset, this bit defaults to zero to enable writing. Setting this bit prevents further writing (excluding the first write that sets this bit).

2.4.3.5 IMP INTERNAL CLOCK SIGNALS. The following paragraphs describe the IMP internal clock signals.

2.4.3.5.1 IMP System Clock. The IMP system clock is supplied to all modules on the IMP (with the exception of the BRG clocks which are connected directly to the VCO output with the PLL enabled). The IMP can be programmed to operate with or without IMP PLL. If IMP PLL is active, the system clock will be driven by PLL clock divider output. If IMP PLL is not active, the system clock will be driven by the PLL input clock (CLKIN).

2.4.3.5.2 BRG Clock. The clock to the BRGs can be supplied from the IMP PLL input (CLKIN) when the IMP PLL is disabled, or from the IMP PLL VCO output (when the PLL is enabled). The BRG prescaler input clock may be optionally programmed to be divided by 2 to allow very low baud rates to be generated from the system clock by setting the BCD bit in the IOMCR.

2.4.3.5.3 PIT Clock. CLKIN is supplied to the periodic interrupt timer (PIT) submodule which allows the PIT clock to run independently of the system clock (refer to Figure 2-2 and Section 3 System Integration Block (SIB)).

2.4.3.6 IMP PLL PINS. The following pins are dedicated to the IMP PLL operation.

2.4.3.6.1 VCCSYN. This pin is the V_{CC} dedicated to the analog IMP PLL circuits. The voltage should be well regulated, and the pin should be provided with an extremely low-impedance path to the V_{CC} power rail if the PLL is to be enabled. VCCSYN should be bypassed to GNDSYN by a 0.1-µF capacitor located as close as possible to the chip package. VCCSYN should be tied to ground if the PLL is to be disabled.

2.4.3.6.2 GNDSYN. This pin is the GND dedicated to the analog IMP PLL circuits. The pin should be provided with an extremely low-impedance path to ground. GDNSYN should be bypassed to VCCSYN by a 0.1 µF capacitor located as close as possible to the chip package. The user should also bypass GNDSYN to VCCSYN with a 0.01 µF capacitor as close as possible to the chip package.

2.4.3.6.3 XFC. This pin connects to the off-chip capacitor for the PLL filter. One terminal of the capacitor is connected to XFC; the other terminal is connected to IQVCC.

2.4.3.6.4 MODCLK. MODCLK specifies what the initial VCO frequency is after a hardware reset if VCCSYN is tied high. During the assertion of RESET, the value of the VCCSYN and MODCLK input pins causes the PEN bit and the MF11–0 bits of the IMP PLL and Clock Control Register (IPLCR) $0F8 to be appropriately written.VCCSYN and MODCLK also determines if the oscillator’s prescaler is used. After RESET is negated, the MODCLK pins is ignored and becomes PA12. Table 2-2 shows the combinations of VCCSYN and MODCLK pins with the corresponding default settings.

2.4.4 IMP Power Management

The IMP portion of the MC68PM302 has several low power modes from which to choose.

2.4.4.1 IMP LOW POWER MODES. The MC68PM302 provides a number of low power modes for the IMP section. Each of the operation modes has different current consumption, wake-up time, and functionality characteristics. The state of the IMP’s 68000 data and address bus lines can be either driven high, low or high impedance during low power stop mode by programming the low power drive control register (LPDCR).

NOTE

> For lowest current consumption, the SCCs and BRGs should be disabled before entering the low power modes. Current consumption for all operating modes is specified in Section 7 Electrical Characteristics.

Table 2-3. IMP Low Power Modes - IMP PLL Enabled

Operation Mode	Oscillator	PLL	Clock	Wake_Up	Current Consumption (Approximate)	Method of Entry/ LPM bits	IMP Functionality
STOP	Not Active	Not active	Not active	70000 osc. clocks	<0.1mA	Stop instruction/ LPM1–0=11	No
DOZE	Active	Not active	Not active	2500 osc. clocks	About 500uA	Stop instruction/ LPM1–0=10	No
STAND_BY	Active	Active (if enabled	Not active	2–5 system clock cycles	About 5mA	Stop instruction/ LPM1–0=01	Partial (BRG clock is active)
SLOW_GO/ NORMAL	Active	Active (if enabled)	Active		Low, depends on CLK freq.	Write to DF3–0	Full

2.4.4.1.1 STOP Mode. In STOP mode, all parts of IMP are inactive and the current consumption is less than 0.1mA. Both the crystal oscillator and the IMP PLL are shut down. Because both the oscillator and the PLL must start up, the wake-up time takes 70000 EXTAL clocks (for example, 70000 cycles of 32.768 kHz crystal will take about 2.2 seconds).

The STOP mode is entered by executing the STOP instruction with the LPM0–1 bits in the IOMCR register set to 11. Refer to 2.4.4.2.2 Entering the STOP/ DOZE/ STAND_BY Mode for an example instruction sequence for use with the STOP instruction.

2.4.4.1.2 DOZE Mode. In DOZE mode, the oscillator is active in the IMP but the IMP PLL is shut down. The current consumption depends on the frequency of the external crystal but is on the order of 500 µa. In DOZE mode, the IMP is shut down. The wake-up time is 2500 cycles of the external crystal (for example, 2500 cycles of 32.768 kHz crystal will take about 80 milliseconds). Doze mode has faster wake-up time than the STOP mode, at the price of higher current consumption.

The DOZE mode is entered by executing the STOP instruction with the LPM1–0 bits in the IOMCR register set to 10. Refer to 2.4.4.2.2 Entering the STOP/ DOZE/ STAND_BY Mode for an example instruction sequence for use with the STOP instruction.

2.4.4.1.3 STAND_BY Mode. In STAND_BY mode, the oscillator is active, and the IMP PLL, if enabled, is active but the IMP clock is not active and the IMP is shut down. Current consumption in STAND-BY mode is less than less than 5mA. The wake up time is a few IMP system clock cycles.

The STAND_BY mode is entered by executing the STOP instruction with the LPM1–0 bits in the IOMCR register set to 01. Refer to 2.4.4.2.2 Entering the STOP/ DOZE/ STAND_BY Mode for an example instruction sequence for use with the STOP instruction.

In STAND-BY mode, the PCMCIA interface is active and any access on the PCMCIA interface will cause the IMP to come out of STAND-BY mode. The STAND_BY mode is useful in applications which have time slots with no activity, and require a minimal wake-up time before IMP returns to NORMAL or SLOW-GO mode.

2.4.4.1.4 SLOW_GO Mode. In the SLOW-GO mode, the IMP is fully operational but the IMP PLL divider has been programmed with a value that is dividing the IMP PLL VCO output to the system clock in order to save power. The PLL output divider can only be used with the IMP PLL enabled. The divider value is programmed in the DF3–0 bits in the IOMCR. The clock may be divided by a power of 2 (2^0 – 2^{15}). No functionality is lost in SLOW-GO mode.

2.4.4.1.5 NORMAL Mode. In NORMAL mode the IMP part is fully operational and the system clock from the PLL is not being divided down.

2.4.4.1.6 IMP Operation Mode Control Register (IOMCR). IOMCR is a 8-bit read/ write register used to control the operation modes of the IMP. The WP bit in IOMCR is used as a protect mechanism to prevent erroneous writing of IOMCR.

IOMCR $0FA

7	6	5	4	3	2	1	0
IOMWP	DF3	DF2	DF1	DF0	BCD	LPM1	LPM0
RESET: 0	0	0	0	0	0	0	0

Read/Write

IOMWP—IMP Operation Mode Control Write Protect Bit

This bit prevents accidental writing into the IOMCR. After reset, this bit defaults to zero to enable writing. Setting this bit prevents further writing (excluding the first write that sets this bit).

DF 3–0—Divide Factor

The Divide Factor Bits define the divide factor of the low power divider of the PLL. These bits specify a divide range between 2^0 and 2^{15}. Changing the value of these bits will not cause a loss of lock condition to the IMP PLL.

BCD—BRG Clock Divide Control

This bit controls whether the divide-by-two block shown in Figure 2-2 is enabled.

0 = The BRG clock is divided by 1.
1 = The BRG clock is divided by 2.

LPM—Low Power Modes

When the 68000 core executes the STOP instruction, the IMP will enter the specified mode.

LPM1–0:

00 = Normal - the IMP PLL and clock oscillator will continue to operate normally.
01 = Stand_by Mode
10 = DOZE Mode
11 = Stop Mode

2.4.4.1.7 Low Power Drive Control Register (LPDCR). This register controls the state of the IMP's 68000 address and data buses during the Standby, Doze, and Stop modes. By programming this register it is possible to minimize power consumption due to external pullups or pull downs, or floating inputs.

LPDCR BAR+$82A

7	6	5	4	3	2	1	0
					LPAL	LPDL	LPDEN
RESET: 0	0	0	0	0	0	0	0

Read/Write

LPDEN—Low Power Drive Enable

0 - The IMP 68000 data and address buses will be high impedance.
1 - The IMP 68000 data and address buses to be driven according to the LPDL bit.

LPDL—Low Power Drive Data Low

0 - The data bus will be driven high when the LPDEN bit is set.
1 - The data bus will be driven low when the LPDEN bit is set.

LPAL—Low Power Drive Address Low

0 - The address bus will be driven high when the LPDEN bit is set.
1 - The address bus will be driven low when the LPDEN bit is set.

NOTE

In 8-bit, non-PCMCIA mode, A0 is also driven during low power. With the PGA package in non-PCMCIA mode, the FC2-FC0 pins are also driven during low power. With the TQFP package in non-PCMCIA mode, the PC_D1–PC_D7 pins are also driven during low power.

2.4.4.1.8 IMP Power Down Register (IPWRD). The IPWRD is a 8-bit read/ write register located at $0FB that is used to control the low power operation of the IMP. This register must be written with the same operand as the STOP instruction that follows. This tells the hardware what level of interrupt (and above) will stop the MC68PM302 from entering low power if it occurs while the clocks are being stopped.

2.4.4.1.9 Default Operation Modes. See 2.4.2.1 Default System Clock Generation.

2.4.4.2 LOW POWER SUPPORT. The following sections describe how to enter the various low power modes.

2.4.4.2.1 Enter the SLOW_GO mode. When the required IMP performance can be achieved with a lower clock rate, the user can reduce power consumption by dividing IMP PLL output clock that provides the IMP system clock. Switching between the NORMAL and SLOW_GO modes is achieved by changing the DF3–0 field in the IOMCR register to a non-zero value. The IMP PLL will not lose lock when the DF3–0 field in the IOMCR register is changed.

2.4.4.2.2 Entering the STOP/ DOZE/ STAND_BY Mode. Entering the STOP/ DOZE/ STAND_BY mode is achieved by the 68000 core executing the following code:

```
nop
move.b     *+6(PC),$000000FB              ;copy STOP operand high byte to addr 000000fb
stop       #$xxxx                         ;xxxx -> SR
nop
```

This code is position independent. The core must be in the supervisor state to execute the STOP instruction, therefore the write to $000000FB must be done in the supervisor state (function code 5, supervisor data). The core trace exception should be disabled, otherwise the low power control will not enter the STOP mode.

To guarantee supervisor state and trace exceptions disabled, this code should be part of a TRAP routine. Upon entering the trap routine, examine the stacked status register. If it indicates the supervisor state, then execute this code to enter STOP mode. If not supervisor, do NOT execute this code (could perform some application-specific error):

```
TRAP_x       btst.b      #5,(SP)                 ; supervisor?
             beq.s       NO_STOP
             nop                                 ; flush execution, bus pipes
             move.b      *+6(PC),$000000FB       ;copy STOP operand high byte to addr 000000fb
             stop        #$xxxx                  ; xxxx -> SR
             nop
             rte
NO_STOP      ...                                 ; error routine?
```

NOTE

> The RI/PB9, DTE/PB10 and PIT conditions will generate a level 4 interrupt. The PCMCIA will generate a level 1, 6, or 7 interrupt. The user should set the 68000 interrupt mask register to the appropriate level before executing this code.

IMP's low power control logic will:

1. Detect the write cycle.
2. Check if bit 5 = 1 (supervisor space) (if it is 0, the low power request will be ignored).
3. Sample the interrupt mask bits (bits 0–2). If during this process of stopping the clocks an interrupt of higher level than the mask is asserted to the core, this process will abort.
4. Wait for 16 clocks to guarantee the execution of the STOP command by the core. $\overline{\text{BG}}$ and $\overline{\text{BGACK}}$ will reset the 16-clock counter and it will restart its count.
5. Assert bus request signal to the core.
6. Wait for Bus Grant from the core
7. Force the IMP to the selected power-down mode, as defined in Table 2-3.

If the IMP is in STAND_BY mode, any PCMCIA access will initiate a wake-up from STAND_BY mode (See All PCMCIA locations (registers, CIS memory, etc.) that can be accessed by the host may also be accessed by the 68000 core. Power Down Options).

NOTE

> The RI (PB9), DTE (PB10) and periodic interrupt timer timeout interrupts conditions will generate level 4 interrupts. The PCM-

> CIA controller can be programmed to generate either level 1, 6, or 7 interrupts. The user should also set the 68000 interrupt mask in the status register (SR) to the appropriate level before executing the STOP command to ensure that the IMP will wake up to the desired events.

2.4.4.2.3 IMP Wake-Up from Low Power STOP Modes. The IMP can wake up from STOP/DOZE/STAND_BY mode to NORMAL/SLOW_GO mode in response to inputs from the following sources:

1. Asserting both $\overline{\text{RESET}}$ and $\overline{\text{HALT}}$ (hard reset) pins.
2. Asserting (high to low transition) either RI (PB9) or DTE (PB10) pins (if these interrupts are enabled).
3. A timeout of the periodic interrupt timer (if the PIT interrupt is enabled).
4. Reset of the PwrDwn bit in the PCMCIA card configuration and status register when the PCMCIA controller is enabled.

When one of these events occur (and the corresponding event bit is set), the IMP low power controller will asynchronously restart the IMP clocks. Then IMP low power control logic will release the 68000 bus and the IMP will return to normal operation. If one of the above wake-up events occurs during the execution of the STOP command, the low power control logic will abort the power down sequence and return to normal operation.

NOTE

> The PB9, PB10, and periodic interrupt timer timeout interrupts conditions will generate level 4 interrupts.The user should also set the 68000 interrupt mask in the status register (SR) to the appropriate level before executing the STOP command to ensure that the IMP will wake up to the desired events.

2.4.4.2.4 IMP Wake-Up Control Register – IWUCR. The IWUCR contains control for the wake-up options. This register can be read and written by the 68000 core.

IWUCR $0F7

7	6	5	4	3	2	1	0
0	PITE	PB10E	PB9Ev	0	PITEn	PB10En	PB9En
RESET: 0	0	0	0	0	0	0	0

Read/Write

PB9Ev—PB9 Event

This bit will be set to one when there is a high to low transition on the PB9 pin. When PB9En is set and PB9Ev is set, the IMP will wake-up from the selected power down state, and a PB9 Interrupt will be generated. The IMP cannot enter the power-down mode if

PB9Ev and PB9En are both set to one. PB9Ev is cleared by writing a one (writing a zero has no effect).

In modem applications $\overline{\mathrm{RI}}$ should be connected to the PB9 pin.

PB10Ev—PB10 Event

This bit will be set to one when there is a high to low transition on the PB10 pin. When PB10En is set and PB10Ev is set, the IMP will wake-up from the selected power down state, and the PB10 Interrupt will be generated. The IMP cannot enter the power-down mode when PB10Ev and PB10En are both set to one. PB10Ev is cleared by writing a one (writing a zero has no effect).

In modem applications the DTE TxD line may be connected to the PB10 pin.

PITEv—PIT Event

This bit will be set to one when there is a time-out on the periodic interrupt timer (PIT). When PITEn bit is set and a time-out occurs (PITEv is set), the IMP will wake-up from the selected power down, and a PIT Interrupt will be generated. The IMP cannot enter the power-down mode if PITEv and PITEn are both set to one. PITEv is cleared by writing a one (writing a zero has no effect).

PB9En—PB9 Enable

This bit, when set, enables the IMP to wake up from power down mode and generate an interrupt when the PB9 Event bit becomes set.

PB10En—PB10 Enable

This bit, when set, enables the IMP to wake up from power down mode and generate an interrupt when the PB10 Event bit becomes set.

PITEn—PIT Enable

This bit, when set, enables the IMP to wake up from power down mode and generate an interrupt when the PIT event bit becomes set, see 3.7.3 Periodic Interrupt Timer (PIT).

2.4.4.3 Fast Wake-Up

In a system clocked with a 32-kHz oscillator, the wake-up recovery time from doze and stop modes may be too long for some applications. In order to shorten this time, an internal ring oscillator (called Ringo) can clock the chip (the term 'real clock' in the following discussion refers to the clock whose source is the external oscillator or crystal; the PLL can be either enabled or disabled). Two reasons for using the fast wake-up are:

1. To allow PC access to the IMP/system bus when there is no need to wake up the real clock (e.g. read status of 16550). In this case, only the internal oscillator is enabled for the duration of the access. The PLL and other clock logic are not awakened.
2. To allow PC accesses and other logic to operate in the time frame between the wake-up command and the actual real clock recovery (from the external crystal or oscillator).

NOTE

If the SCCs use the internal clock, or if they use external clock and the Ringo/external frequency ratio does not comply with the

1 / 2.5 maximum ration specification, then they cannot be enabled until the real clock has resumed operation.

The criteria for enabling Ringo and waking up the CPU (by giving it an interrupt) are:

— RINGOEN=1 and an unmasked event in the PCMCIA access wake-up event register (PCAWER) occurs.
— RINGOEN=1 and an unmasked PITEv, PB10Ev, or PB9Ev event occurs (please refer to the IWUCR register)

The internal ring oscillator is not enabled if the PLL is disabled. A system with the PLL disabled has to have an external oscillator connected in order to shorten the wake-up time.

There are two possible interrupts to the CPU from the Ringo logic:

— Interrupt when Ringo is enabled; the CPU is always interrupted when Ringo starts oscillating (RINGOEN bit enables both ring oscillator and enables the interrupt to the CPU). Event bits for this interrupt are all wake-up events.
— Maskable interrupt when the system clock switches to the real clock. The event bit for this interrupt is in the ring oscillator event register.

The Ringo interrupts can be either at level 1, 6, or 7 according to RICR bits. If the CPU determines that the system needs the real clock, it programs the RECLMODE bits which enables the oscillator and PLL and interrupts if necessary; if it decides that the system can go back to sleep, it executes the normal power-down sequence which turns off Ringo. Upon switching to the real clock, the CPU can be interrupted by programming the RICR bits. (Note that Ringo is turned off either at the end of a power down sequence, or when the PLL has gained lock). If the RECLMODE bits are programmed to enable the PLL and the oscillator, the user is allowed to enter low power mode *after* the real clock has resumed. The chip will not operate correctly if the CPU enters the low power sequence while the PLL is waking up. Resetting of the RINGOEN bit is allowed only if the system is clocked by the real clock. The CLKO signal can be disabled by software if the user cannot operate the system at the Ringo frequency.

2.4.4.3.5 Ring Oscillator Control Register (RINGOCR)

RINGOCR BAR+$81A

7	6	5	4	3	2	1	0
			RICR		RECLMODE		RINGOEN
RESET: 0	0	0	0	0	0	0	0

Read/Write

RINGOEN — Ring Oscillator Enable

0 = Ring oscillator is not used
1 = Ring oscillator is enabled

RECLMODE — Real Clock Mode

00 = Do not enable the real clock
01 = Enable the real clock and switch the system clock from Ringo to the real clock once it is stable
10 = Enable the real clock and generate an interrupt to the CPU after the switch occurs
11 = Reserved

RICR — Ring Oscillator Interrupt Control

00 = Connect Ringo Interrupts to 68k interrupt request level 1
01 = Connect Ringo Interrupts to 68k Interrupt request level 6
10 = Connect Ringo Interrupts to 68k interrupt request level 7
11 = Reserved

2.4.4.3.6 Ring Oscillator Event Register (RINGOEVR).

RINGOEVR BAR+$81B

7	6	5	4	3	2	1	0
RESERVED							RECLSEV
RESET:							
0	0	0	0	0	0	0	0

Read/Write

RECLSEV — Real Clock Switch Event

0 = Event has not occurred
1 = Real clock is now the system clock (This bit is reset by writing 1)

Bits 7-1 — Reserved

2.5 MC68PM302 DUAL PORT RAM

The internal 1152-byte dual-port RAM has 576 bytes of system RAM (see Table 2-4) and 576 bytes of parameter RAM (see Table 2-5).

Table 2-4. System RAM

Address	Width	Block	Description
Base + 000 • • • Base + 23F	576 Bytes	RAM	User Data Memory
Base +240 • • • Base + 3FF			Reserved (Not Implemented)

The parameter RAM contains the buffer descriptors for each of the three SCC channels, the 16550 channel or SCC2, the SCP, and the two SMC channels. The memory structures of the three SCC channels are identical. When any SCC, SCP, or SMC channel buffer descriptors or parameters are not used, their parameter RAM area can be used for additional memory. For detailed information about the use of the buffer descriptors and protocol parameters in a specific protocol, see Section 4 CP. Base + 67E contains the MC68PM302 revision number.

Table 2-5. Parameter RAM

Address	Width	Block	Description
Base + 400	4 Word	SCC1	Rx BD 0
Base + 408	4 Word	SCC1	Rx BD 1
Base + 410	4 Word	SCC1	Rx BD 2
Base + 418	4 Word	SCC1	Rx BD 3
Base + 420	4 Word	SCC1	Rx BD 4
Base + 428	4 Word	SCC1	Rx BD 5
Base + 430	4 Word	SCC1	Rx BD 6
Base + 438	4 Word	SCC1	Rx BD 7
Base + 440	4 Word	SCC1	Tx BD 0
Base + 448	4 Word	SCC1	Tx BD 1
Base + 450	4 Word	SCC1	Tx BD 2
Base + 458	4 Word	SCC1	Tx BD 3
Base + 460	4 Word	SCC1	Tx BD 4
Base + 468	4 Word	SCC1	Tx BD 5
Base + 470	4 Word	SCC1	Tx BD 6
Base + 478	4 Word	SCC1	Tx BD 7
Base + 480 • • • Base + 4BF		SCC1 SCC1	Specific Protocol Parameters
Base + 4C0 • • • Base + 4FF			Reserved (Not Implemented)
Base + 500	4 Word	SCC2, 16550	Rx BD 0
Base + 508	4 Word	SCC2, 16550	Rx BD 1
Base + 510	4 Word	SCC2, 16550	Rx BD 2
Base + 518	4 Word	SCC2, 16550	Rx BD 3
Base + 520	4 Word	SCC2, 16550	Rx BD 4
Base + 528	4 Word	SCC2, 16550	Rx BD 5
Base + 530	4 Word	SCC2, 16550	Rx BD 6
Base + 538	4 Word	SCC2, 16550	Rx BD 7
Base + 540	4 Word	SCC2, 16550	Tx BD 0
Base + 548	4 Word	SCC2, 16550	Tx BD 1
Base + 550	4 Word	SCC2, 16550	Tx BD 2
Base + 558	4 Word	SCC2, 16550	Tx BD 3
Base + 560	4 Word	SCC2, 16550	Tx BD 4
Base + 568	4 Word	SCC2, 16550	Tx BD 5
Base + 570	4 Word	SCC2, 16550	Tx BD 6/DRAM Refresh
Base + 578	4 Word	SCC2, 16550	Tx BD 7/DRAM Refresh
Base + 580 • • • Base + 5BF		SCC2 SCC2	Specific Protocol Parameters
Base + 5C0 • • • Base + 5FF			Reserved (Not Implemented)
Base + 600	4 Word	SCC3	Rx BD 0
Base + 608	4 Word	SCC3	Rx BD 1
Base + 610	4 Word	SCC3	Rx BD 2
Base + 618	4 Word	SCC3	Rx BD 3
Base + 620	4 Word	SCC3	Rx BD 4
Base + 628	4 Word	SCC3	Rx BD 5
Base + 630	4 Word	SCC3	Rx BD 6
Base + 638	4 Word	SCC3	Rx BD 7
Base + 640	4 Word	SCC3	Tx BD 0
Base + 648	4 Word	SCC3	Tx BD 1
Base + 650	4 Word	SCC3	Tx BD 2
Base + 658	4 Word	SCC3	Tx BD 3 ##
Base + 660	3 Word	SMC	Reserved
Base + 666	Word	SMC1	Rx BD
Base + 668	Word	SMC1	Tx BD
Base + 66A	Word	SMC2	Rx BD
Base + 66C	Word	SMC2	Tx BD
Base + 66E #	6 Word	SMC1–SMC2	Internal Use
Base + 67A	Word	SCP	Rx/Tx BD
Base + 67C	Word	SCC1–SCC3	BERR Channel Number
Base + 67E #	Word	CP	MC68PM302 Revision Number

Table 2-5. Parameter RAM

Base + 680 • • • Base + 6BF		SCC3 SCC3	Specific Protocol Parameters
Base + 6C0 • • • Base + 7FF			Reserved (Not Implemented)

# Modified by the CP after a CP or system reset.
## Tx BD 4, 5, 6, and 7 are not initially available to SCC3. (4.3.5 Buffer Descriptors Table for information on how they may be regained.)

In addition to the internal dual-port RAM, a number of internal registers support the functions of the various M68000 core peripherals. The internal registers (see Table 2-6) are memory-mapped registers offset from the BAR pointer and are located on the internal M68000 bus.

NOTE

All undefined and reserved bits within registers and parameter RAM values written by the user in a given application should be written with zero to allow for future enhancements to the device.

2.6 INTERNAL REGISTERS MAP

Table 2-6. Internal Registers Map

Address	Name	Width	Block	Description	Reset Value
Base + 800	RES	16	IDMA	Reserved	
Base + 802	CMR	16	IDMA	Channel Mode Register	0000
Base + 804	SAPR	32	IDMA	Source Address Pointer	XXXX XXXX
Base + 808	DAPR	32	IDMA	Destination Address Pointer	XXXX XXXX
Base + 80C	BCR	16	IDMA	Byte Count Register	00
Base + 80E *	CSR	8	IDMA	Channel Status Register	
Base + 80F	RES	8	IDMA	Reserved	
Base + 810	FCR	8	IDMA	Function Code Register	XX
Base + 811	RES	8	IDMA	Reserved	
Base + 812 #	GIMR	16	Int Cont	Global Interrupt Mode Register	0000
Base + 814 *	IPR	16	Int Cont	Interrupt Pending Register	0000
Base + 816	IMR	16	Int Cont	Interrupt Mask Register	0000
Base + 818 *	ISR	16	Int Cont	In-Service Register	0000
Base + 81A	RINGOCR	8	Low Power	RINGO Control Register	00
Base + 81B*	RINGOEVR	8	Low Power	RINGO Event Register	00
Base + 81C	RES	16	Int Cont	Reserved	
Base + 81E #	PACNT	16	PIO	Port A Control Register	0000
Base + 820 #	PADDR	16	PIO	Port A Data Direction Register	0000
Base + 822 #	PADAT	16	PIO	Port A Data Register	XXXX ##
Base + 824 #	PBCNT	16	PIO	Port B Control Register	0080
Base + 826 #	PBDDR	16	PIO	Port B Data Direction Register	0000
Base + 828 #	PBDAT	16	PIO	Port B Data Register	XXXX ##
Base + 82A	LPDCR	8	PIO	Low Power Drive Control Register	00
Base + 82C	PPR	8	CS	PCMCIA protection register	0000
Base + 82E	RES	16	CS	Reserved	
Base + 830 #	BR0	16	CS0	Base Register 0	C001
Base + 832 #	OR0	16	CS0	Option Register 0	DFFD
Base + 834 #	BR1	16	CS1	Base Register 1	C000
Base + 836 #	OR1	16	CS1	Option Register 1	DFFD
Base + 838 #	BR2	16	CS2	Base Register 2	C000
Base + 83A #	OR2	16	CS2	Option Register 2	DFFD
Base + 83C #	BR3	16	CS3	Base Register 3	C000
Base + 83E #	OR3	16	CS3	Option Register 3	DFFD
Base + 840	TMR1	16	Timer	Timer Unit 1 Mode Register	0000
Base + 842	TRR1	16	Timer	Timer Unit 1 Reference Register	FFFF
Base + 844	TCR1	16	Timer	Timer Unit 1 Capture Register	0000
Base + 846	TCN1	16	Timer	Timer Unit 1 Counter	0000
Base + 848	RES	8	Timer	Reserved	
Base + 849 *	TER1	8	Timer	Timer Unit 1 Event Register	00
Base + 84A	WRR	16	WD	Watchdog Reference Register	FFFF
Base + 84C	WCN	16	WD	Watchdog Counter	0000
Base + 84E	RES	16	Timer	Reserved	
Base + 850	TMR2	16	Timer	Timer Unit 2 Mode Register	0000
Base + 852	TRR2	16	Timer	Timer Unit 2 Reference Register	FFFF
Base + 854	TCR2	16	Timer	Timer Unit 2 Capture Register	0000
Base + 856	TCN2	16	Timer	Timer Unit 2 Counter	0000
Base + 858	RES	8	Timer	Reserved	

Table 2-6. Internal Registers Map

Address	Name	Width	Block	Description	Reset Value
Base + 859 *	TER2	8	Timer	Timer Unit 2 Event Register	00
Base + 85A	RES	16	Timer	Reserved	
Base + 85C	RES	16	Timer	Reserved	
Base + 85E	RES	16	Timer	Reserved	
Base + 860	CR	8	CP	Command Register	00
Base + 861 Base + 86F	Reserved				
Base + 870	COR	8	PC Cont	Configuration Option Register	00
Base + 871	CCSR	8	PC Cont	Card Configuration and Status Register	00
Base + 872	PRR	8	PC Cont	Pin Replacement Register	00
Base + 873	PSCR	8	PC Cont	Socket and Copy Register	XX
Base + 874	IOEIR	8	PC Cont	I/O Event Indication register	00
Base + 875	RES1	8	PC Cont	Reserved1	00
Base + 876	RES2	8	PC Cont	Reserved2	00
Base + 877	RES3	8	PC Cont	Reserved3	00
Base + 878	PCDMAD	16	PC DMA	PC DMA Data Register	
Base + 87a Base + 87F	Reserved				
Base + 880	RES	16	SCC1	Reserved	
Base + 882	SCON1	16	SCC1	SCC1 Configuration Register	0004
Base + 884	SCM1	16	SCC1	SCC1 Mode Register	0000
Base + 886	DSR1	16	SCC1	SCC1 Data Sync. Register	7E7E
Base + 888 *	SCCE1	8	SCC1	SCC1 Event Register	00
Base + 889	RES	8	SCC1	Reserved	
Base + 88A	SCCM1	8	SCC1	SCC1 Mask Register	00
Base + 88B	RES	8	SCC1	Reserved	
Base + 88C	SCCS1	8	SCC1	SCC1 Status Register	00
Base + 88D	RES	8	SCC1	Reserved	
Base + 88E	RES	16	SCC1	Reserved	
Base + 890	RES	16	SCC2	Reserved	
Base + 892	SCON2	16	SCC2	SCC2 Configuration Register	0004
Base + 894	SCM2	16	SCC2	SCC2 Mode Register	0000
Base + 896	DSR2	16	SCC2 16550	SCC2 Data Sync. Register	7E7E
Base + 898 *	SCCE2, E550	16	SCC2 16550	SCC2,16550 Emulation Event Register	0000
Base + 899	RES	8	SCC2	Reserved	
Base + 89A	SCCM2, M550	8	SCC2 16550	SCC2,16550 Emulation Mask Register	00
Base + 89B	RES	8	SCC2	Reserved	
Base + 89C	SCCS2, S550	8	SCC2 16550	SCC2,16550 Emulation Status Register	0000
Base + 89D	RES	8	SCC2	Reserved	
Base + 89E	RES	16	SCC2	Reserved	
Base + 8A0	RES	16	SCC3	Reserved	
Base + 8A2	SCON3	16	SCC3	SCC3 Configuration Register	0004
Base + 8A4	SCM3	16	SCC3	SCC3 Mode Register	0000
Base + 8A6	DSR3	16	SCC3	SCC3 Data Sync. Register	7E7E
Base + 8A8 *	SCCE3	8	SCC3	SCC3 Event Register	00

Table 2-6. Internal Registers Map

Address	Name	Width	Block	Description	Reset Value
Base + 8A9	RES	8	SCC3	Reserved	
Base + 8AA	SCCM3	8	SCC3	SCC3 Mask Register	00
Base + 8AB	RES	8	SCC3	Reserved	
Base + 8AC	SCCS3	8	SCC3	SCC3 Status Register	00
Base + 8AD	RES	8	SCC3	Reserved	
Base + 8AE	RES	16	SCC3	Reserved	
Base + 8B0	SPMODE	16	SCM	SCP, SMC Mode and Clock Control Register	0000
Base + 8B2 #	SIMASK	16	SI	Serial Interface Mask Register	FFFF
Base + 8B4 #	SIMODE	16	SI	Serial Interface Mode Register	0000
Base + 8B6 Base +8BF	Reserved				
Base + 8C0	PCMR	24	PCMCIA	PCMCIA Mode Register	000000
Base + 8C4 *****	PCRWER	8	PCMCIA	PCMCIA Configuration Registers Write Event Register	00
Base + 8C5 *****	PCAWER	8	PCMCIA	PCMCIA Access Wake-up Event Register	00
Base + 8C6	PCRWMR	8	PCMCIA	PCMCIA Configuration Registers Write Mask Register	00
Base + 8C7	PCAWMR	8	PCMCIA	PCMCIA Access Wake-up Mask Register	00
Base + 8C8 *****	PCHER	8	PCMCIA	PCMCIA Host (PC) Event Register	00
Base + 8CC	CISBAR	16	PCMCIA	CIS Base Address Register	0000
Base + 8D0	CMBAR1	16	PCMCIA	Common Memory Space Base Address Register1	0000
Base + 8D4	CMBAR2	16	PCMCIA	Common Memory Space Base Address Register2	0000
Base + 8D8 Base + 8DF	Reserved				
Base + 8E0	PUCR	8		Pull-Up Control Register	00
Base + 8E2	RES	16		Reserved	
Base + 8E4	RES	16		Reserved	
Base + 8E6	PDDAT	16		Port D Data Register	XXXX*
Base + 8E8 Base + 8ED	Reserved				
Base + 8EE	BRGP	8		Baud Rate Generator Pins Register	0
Base + 8EF	Reserve d				
Base + 8F0	EMR	16	16550	16550 Emulation Mode Register	0000
Base + 8F2	LCR	8	16550	16550 Emulation Line Control Register	00
Base + 8F3	MSR	8	16550	16550 Emulation Modem Status Register	00
Base + 8F4	DLM	8	16550	16550 Emulation Divisor Latch (MSB)	XX
Base + 8F5	DLL	8	16550	16550 Emulation Divisor Latch (LSB)	XX
Base + 8F6	LSR	8	16550	16550 Emulation Line Status Register	00
Base + 8F7 Base + FFF				Reserved (Not implemented)	

# Reset only upon total system reset. (RESET and HALT assert together), but not on the execution of an M68000 RESET instruction. See the RESET pin description for details.

## The output latches are undefined at total system reset.

***** Event register with special properties.

SECTION 3
SYSTEM INTEGRATION BLOCK (SIB)

The MC68PM302 contains an extensive SIB that simplifies the job of both the hardware and software designer. Most of the features are taken from the MC68302 without change, features that have been added or changed are highlighted in **bold** text.

NOTE

> This section will only present the register descriptions for each block. For more information on the operation of each block, please refer to the *MC68302 Users' Manual*. Items that are new or have changed will be described in detail.

The SIB includes the following functions:

- IDMA Controller with Three Handshake Signals: DREQ, DACK, and DONE
- Interrupt Controller with Two Modes of Operation
- **More** Parallel Input/Output (I/O) Ports, Some with Interrupt Capability
- **Pullup Resistors on Parallel Input and PCMCIA Input Pins**
- On-Chip 1152-Byte Dual-Port RAM
- Four Timers Including a Watchdog Timer and **Periodic Interrupt Timer**
- Four Programmable Chip-Select Lines with Wait-State Generator Logic
- **Glueless Interface to SRAM, EPROM, Flash EPROM, and EEPROM**
- System Control
  - —System Status and Control Logic
  - —Disable CPU Logic (M68000)
  - —Bus Arbitration Logic with Low-Interrupt Latency Support
  - —Hardware Watchdog for Monitoring Bus Activity
  - —DRAM Refresh Controller
  - **—Programmable Bus Width**
- **Boot from SCC**

3.1 SYSTEM CONTROL

The IMP system control consists of a System Control Register (SCR) that configures the following functions:

- System Status and Control Logic

- $\overline{\text{AS}}$ Control During Read-Modify-Write-Cycles
- Disable CPU (M68000) Logic
- Bus Arbitration Logic with Low-Interrupt Latency Support
- Hardware Watchdog
- Low-Power (Standby) Modes
- Freeze Control

3.1.1 System Control Register (SCR)

The SCR is a 32-bit register that consists of system status and control bits, a bus arbiter control bit, and hardware watchdog control bits. Refer to Figure 3-1 and to the following paragraphs for a description of each bit in this register. The SCR is a memory-mapped read-write register. The address of this register is fixed at $0F4 in supervisor data space (FC = 5).

$F4

31	30	29	28	27	26	25	24
EMEN	0	0	0	IPA	HWT	WPV	ADC

$F5

23	22	21	20	19	18	17	16
RME	ERRE	VGE	WPVE	RMCST	EMWS	ADCE	BCLM

$F6

15	14	13	12	11	10	9	8
FRZW	FRZ2	FRZ1	SAM	HWDEN	HWDCN2–HWDCN0		

Figure 3-1. System Control Register

Table 3-1. SCR Register Bits

Bit	Name
EMEN	External Master Cycle Disabled
IPA	Interrupt Priority Active
HWT	Hardware Watchdog Timeout
WPV	Write Protect Violation
ADC	Address Decode Conflict
RME	Ram Microcode Enable
ERRE	External RISC Request Enable
VGE	Vector Generation Enable
WPVE	Write Protect Violation Enable
RMCST	Read-Modify-Write Cycle Special Treatment
EMWS	External Master Wait State
ADCE	Address Decode Conflict Enable
BCLM	Bus Clear Mask
FRZW	Freeze Watch Dog Timer Enable
FRZ1	Freeze Timer 1 Enable
FRZ2	Freeze Timer 2 Enable
SAM	Synchronous Access Mode
HWDEN	Hardware Watchdog Enable
HWDCN	Hardware Watchdog Count

3.1.2 System Status Bits

Bits 27-24 of the SCR are used to report events recognized by the system control logic. On recognition of an event, this logic sets the corresponding bit in the SCR. The bits may be read at any time. A bit is reset by one and is left unchanged by zero. More than one bit may be reset at a time. For more information on these bits, please refer to the *MC68302 Users' Manual*.

IPA—Interrupt Priority Active

This bit is set when the M68000 core has an unmasked interrupt request.

NOTE

> If BCLM is set, an interrupt handler will normally clear IPA at the end of the interrupt routine to allow an alternate bus master to regain the bus; however, if BCLM is cleared, no additional action need be taken in the interrupt handler.

HWT—Hardware Watchdog Timeout

This bit is set when the hardware watchdog (see 3.8.3 Bus Arbitration Logic) reaches the end of its time interval; $\overline{\text{BERR}}$ is generated following the watchdog timeout, even if this bit is already set.

WPV—Write Protect Violation

This bit is set when a bus master attempts to write to a location that has RW set to zero (read only) in its associated base register (BR3–BR0).

ADC—Address Decode Conflict

This bit is set when a conflict has occurred in the chip-select logic because two or more chip-select lines attempt assertion in the same bus cycle.

3.1.3 System Control Bits

The system control logic uses seven control bits in the SCR.

EMEN—External Master Cycle Using Halt Enable

0 = External master cycles using the $\overline{\text{HALT}}$ Arbitration are disabled.
1 = External master cycles using the $\overline{\text{HALT}}$ Arbitration are enabled.

After system reset, this bit defaults to zero. This bit is used to enable the bus arbitration using $\overline{\text{HALT}}$ when the CPU is enabled and the PCMCIA interface is enabled.

WPVE—Write Protect Violation Enable

0 = $\overline{\text{BERR}}$ is not asserted when a write protect violation occurs.
1 = $\overline{\text{BERR}}$ is asserted when a write protect violation occurs.

RMCST—RMC Cycle Special Treatment

0 = The locked read-modify-write cycles of the TAS instruction will be identical to the M68000 ($\overline{\text{AS}}$ and $\overline{\text{CS}}$ will be asserted during the entire cycle). The arbiter will issue $\overline{\text{BG}}$, regardless of the internal M68000 core $\overline{\text{RMC}}$. If an IMP chip select is used, the $\overline{\text{DTACK}}$ generator will insert wait states on the read cycle only.
1 = The IMP uses the internal $\overline{\text{RMC}}$ to negate $\overline{\text{AS}}$ and $\overline{\text{CS}}$ at the end of the read portion of the RMC cycle and reasserts $\overline{\text{AS}}$ and $\overline{\text{CS}}$ at the beginning of the write portion. $\overline{\text{BG}}$ will not be asserted until the end of the write portion. If an IMP chip select is

used, the $\overline{\text{DTACK}}$ generator will insert wait states on both the read and write portion of the cycles.

The assertion of the internal $\overline{\text{RMC}}$ by the M68000 core is seen by the arbiter and will prevent the arbiter from issuing bus grants until the completion of M68000-initiated locked read-modify-write activity. After system reset, this bit defaults to zero.

EMWS—External Master Wait State (EMWS) **(Valid only in Disable CPU Mode)**

When EMWS is set and an external master is using the chip-select logic for $\overline{\text{DTACK}}$ generation or is synchronously reading from the internal peripherals (SAM = 1), one additional wait state will be inserted in every memory cycle to external memory and peripherals and also in every cycle to internal memory and peripherals. When EMWS is cleared, all synchronous internal accesses will be with zero wait states, and the chip-select logic will generate $\overline{\text{DTACK}}$ after the exact programmed number of wait states. The chip-select lines are asserted slightly earlier for internal master memory cycles than for an external master. EMWS should be set whenever these timing differences will necessitate an additional wait state for external masters. After system reset, this bit defaults to zero.

ADCE—Address Decode Conflict Enable

0 = $\overline{\text{BERR}}$ is not asserted by a conflict in the chip-select logic when two or more chip-select lines are programmed to overlap the same area.

1 = $\overline{\text{BERR}}$ is asserted by a conflict in the chip-select logic when two or more chip-select lines are programmed to overlap the same area.

BCLM—Bus Clear Mask

0 = The arbiter does not use the M68000 core internal IPEND signal to assert the internal and external bus clear signals.

1 = The arbiter uses the M68000 core internal IPEND signal to assert the internal and external bus clear signals.

SAM—Synchronous Access Mode **(Valid only in Disable CPU Mode)**

This bit controls how external masters may access the IMP peripheral area. This bit is not relevant for applications that do not have external bus masters that access the IMP. In applications such as disable CPU mode, in which the M68000 core is not operating, the user should note that SAM may be changed by an external master on the first access of the IMP, but that first write access must be asynchronous with three wait states. (If $\overline{\text{DTACK}}$ is used to terminate bus cycles, this change need not influence hardware.)

0 = Asynchronous accesses. All accesses to the IMP internal RAM and registers (including BAR and SCR) by an external master are asynchronous to the IMP clock. Read and write accesses are with three wait states, and $\overline{\text{DTACK}}$ is asserted by the IMP assuming three wait-state accesses. This is the default value.

1 = Synchronous accesses. All accesses to the IMP internal RAM and registers (including BAR and SCR) must be synchronous to the IMP clock. Synchronous read accesses may occur with one wait state if EMWS is also set to one.

RME—RAM Microcode Enable

This bit is used to initiate the execution of Communication Processor microcode that has been loaded into the dual port RAM.

3.1.4 Hardware Watchdog

The hardware watchdog logic is used to assert an internal $\overline{\text{BERR}}$ ($\overline{\text{BERR}}$ is driven externally wi the PCMCIA interface is disabled) and set HWT when a bus cycle is not terminated by $\overline{\text{DTACK}}$ and after a programmable number of clock cycles has elapsed.

The hardware watchdog logic uses four bits in the SCR.

HWDEN—Hardware Watchdog Enable

0 = The hardware watchdog is disabled.
1 = The hardware watchdog is enabled.

After system reset, this bit defaults to one to enable the hardware watchdog.

HWDCN–HWDCN0—Hardware Watchdog Count 2–0

000 = $\overline{\text{BERR}}$ is asserted after 128 clock cycles (8 µs, 16-MHz clock)
001 = $\overline{\text{BERR}}$ is asserted after 256 clock cycles (16 µs, 16-MHz clock)
010 = $\overline{\text{BERR}}$ is asserted after 512 clock cycles (32 µs, 16-MHz clock)
011 = $\overline{\text{BERR}}$ is asserted after 1K clock cycles (64 µs, 16-MHz clock)
100 = $\overline{\text{BERR}}$ is asserted after 2K clock cycles (128 µs, 16-MHz clock)
101 = $\overline{\text{BERR}}$ is asserted after 4K clock cycles (256 µs, 16-MHz clock)
110 = $\overline{\text{BERR}}$ is asserted after 8K clock cycles (512 µs, 16-MHz clock)
111 = $\overline{\text{BERR}}$ is asserted after 16K clock cycles (1 ms, 16-MHz clock)

After system reset, these bits default to all ones; thus, $\overline{\text{BERR}}$ will be asserted after 1 ms for a 16-MHz system clock.

3.1.5 Freeze Control

Used to freeze the activity of selected peripherals, $\overline{\text{FRZ}}$ is useful for system debugging purposes. When $\overline{\text{FRZ}}$ is asserted (For more information on these bits, please refer to the *MC68302 Users' Manual)*:

FRZ1 — Freeze Timer 1 Enable

0 = Freeze Timer 1 Logic is disabled
1 = Freeze Timer 1 Logic is enabled
After system reset this bit defaults to zero.

FRZ2 — Freeze Timer 2 Enable

0 = Freeze Timer 2Logic is disabled
1 = Freeze Timer 2Logic is enabled

After system reset, this bit defaults to zero.

FRZW — Freeze Watchdog Timer Enable

0 = Freeze Watchdog Timer Logic is disabled
1 = Freeze Watchdog Timer Logic is enabled
After system reset this bit defaults to zero.

3.2 PROGRAMMABLE DATA BUS SIZE SWITCH

The following procedure allows MC68PM302 to be booted in an 8 or 16 bit bus width and then switched to 16 or 8 bit bus width for future accesses. It does not implement true dynamic bus sizing, but allows a software reconfiguration of the BUSW pin.

3.2.1 Enabling the Dynamic Bus Switch

In order to enable the dynamic bus switch, the user must write the BSW bit in the PCMCIA Protection Register (See Section Section 3 System Integration Block (SIB)) and toggling the BSWEN bit in the PPR.

3.2.2 Basic Procedure:

The MC68PM302 is booted in its 8-bit mode by externally connecting the BUSW pin to GND. It is expected that the MC68PM302 will be executing out of EPROM or Flash at this time, and that no external data memory is available in 8-bit mode.

The MC68PM302 initializes the BAR register to place the 4K block of dual-port RAM and peripherals in an area that does not overlap the EPROM region. Note that this is part of a normal 302 initialization sequence. Also note that the CFC bit of the BAR register should NOT be set -- it must be cleared.

At this time other desired initialization should be completed on the MC68PM302. No bus masters (IDMA, SDMA, or external) should be enabled.

While in 8-bit mode, the MC68PM302 should initialize the external memory registers that control the16-bit external memory space. External memory refresh is not enabled at this time, but all other desired external memory control features should be enabled. Note that the MC68PM302 does not access the external memory itself yet, only the external memory control registers.

The MC68PM302 now copies a special boot code to the user area of the internal dual-port RAM and then jumps to the start of that code. This code is copied as “data” to the dual-port RAM.

To summarize, the procedure is then:

- Boot up MC68PM302
- Perform required 8 bit operations
- Write code to the Dual Port RAM for the bus width change
- Jump to code in the dual port RAM

- When ready to use 16 bit bus width, set the BUSW bit to 1
- Toggle the BWSEN bit from zero to one.
- Allow time for bus arbitration and the instruction pipeline to clear
- Initialize external memory
- Copy boot code from EPROM to external memory space
- Execute code from external memory space

NOTE

> The stack which is shared by both codes should be placed in the dual ported RAM. Copy the stack to dual port RAM after switching to the second RAM and change the stack pointer.

3.3 LOAD BOOT CODE FROM AN SCC

The MC68PM302 provides the capability of downloading program code into SCC1 and beginning program execution in the dual port RAM. The boot function has two clocking options: external and internal.

In the first mode, the user provides the chip with an external clock 16* the desired baud rate. In the second mode, the RISC processor programs the SCC into UART mode running at approximately 9600 baud (assuming the frequency of the clock to the chip has one of two nominal values 32.768 Khz or 4.192 Mhz).

The first 576 bytes that are received into SCC1 are stored in the dual-port RAM. No error checking is performed on the incoming serial bit stream. The 68000 processor then begins executing from the first location of the dual-port RAM to complete the boot process. This function is not supported for SCC2 or SCC3.

Three pins are sampled to determine the mode of operation and clock of the boot function:

PA7–Sampled during Hard Reset ($\overline{\text{RESET}}$ and $\overline{\text{HALT}}$ asserted)

0 Boot from SCC is enabled
1 Boot from SCC is disabled

PA5–Sampled within 100 clocks from the negation of $\overline{\text{RESET}}$

0 Internal Clock
1 External Clock 16* the bit rate on TCLK1 and RCLK1

PA12 (MODCLK0) Sampled during Hard Reset ($\overline{\text{RESET}}$ and $\overline{\text{HALT}}$ asserted)

0 Nominal input frequency on EXTAL is 4.192 Mhz
1 Nominal input frequency on EXTAL is 32.768 Khz

To enable the boot function, the PA7 pin must be pulled low during system reset. (System reset is defined by the $\overline{\text{RESET}}$ and $\overline{\text{HALT}}$ pins being asserted.) The PA7 pin must be pulled high during system reset, if boot mode is not to be enabled. Once the MC68PM302 detects that the PA7 pin is asserted, it internally keeps the $\overline{\text{HALT}}$ signal to the 68K core asserted after system reset is complete. This action prevents the 68000 from fetching the reset vector.

NOTE

> PA7 needs to be either pulled UP or pulled DOWN. Do not leave this pin floating during reset.

Once system reset is complete, the RISC processor programs the BAR register to $0000 to place the dual-port RAM at the low end of system memory. It then samples the PA5 pin to determine the clock source for the UART.

NOTE

> PA5 is expected to be valid for 100 clocks after the negation of $\overline{\text{RESET}}$.

If PA5 is pulled high, SCC1 is programmed for external clocks. In this mode, the user has to connect an external clock 16* the bit rate to TCLK1 and RCLK1.

If PA5 is pulled low, the SCC is programmed for internal clocks and the TCLK1 and RCLK1 pins are programmed to three-state to avoid contention with user clocks. The RISC processor then programs the SCON register of the SCC based on PA12. The PA12 value sampled during reset (MODCLK0) is decoded in order to provide ~9600 bps with two input frequencies (4.192 Mhz and 32.768 Khz).

- If MODCLK0 = GND, SCON1 is programmed to 0x00D8.
- If MODCLK0 = VCC, SCON1 is programmed to 0x00A8.

The following baud rates are achieved as a function of VCCSYN, MODCLK, and input clock:

VCCSYN-MODCLK		
00	20 Mhz	11467 bps (SCON1=0x00D8)
10	4.192 Mhz	9614 bps (SCON1=0x00D8)
10	4.8 Mhz	11009 bps (SCON1=0x00D8)
11	32.768 Khz	9662 bps (SCON1=0x00A8)

The following paragraphs explain the boot process for the 32.768 Khz case in detail. If the clock provided to the MC68PM302 is 32.768 KHz, the system frequency is multiplied by 401 to get 13.139968 MHz. The CD10-CD0 bits of the SCON are programmed to 84 decimal giving a UART frequency of 9662. In summary, 13.139968 MHz / (84 + 1) / 16 = 9662. If the starting frequency is exactly 32.000 KHz, the UART frequency is 9435.

NOTE

> The autobaud function cannot be used in the boot download process.
>
> Values in bit CD10:0 in SCON are not relevant if PA5=1

The RISC processor then programs the SCM register to $013D to program the SCC to UART mode with both the receiver and the transmitter enabled, software operation mode ($\overline{\text{CD}}$ and $\overline{\text{CTS}}$ are don't cares), 8-bit data characters, and no parity. The RISC processor then begins receiving data into the dual-port RAM beginning with location $0 of the dual-port RAM. Every character that is received is “echoed” back out of the TXD1 pin. The MC68PM302 UART must be sent 576 bytes of data from the external UART since the MC68PM302 will not leave the boot mode until 576 bytes are received. If the boot program is less than 576 bytes, the user is suggested to write $00 into the remaining locations.

After 576 bytes are received, the RISC programs the SCM register to $0, which clears the ENR and ENT bits to disable the UART (returns to its reset value).

The RISC processor next negates the $\overline{\text{HALT}}$ signal to the core internally. The 68000 then reads the reset vector from the first location of the dual-port RAM. In most cases, the code that is downloaded will enable the chip selects of the MC68PM302, initialize the MC68PM302 receive buffer descriptors of SCC1 to continue receiving additional boot code into external system RAM, and re-initialize the UART receiver.

NOTE

The first 576 bytes also overlays the exception vector table, meaning that exception vectors will not work unless the user carefully maps the code around certain desired vectors and points those vectors into the 576 byte code space. In addition the stack pointer must point into the 576 bytes if any exceptions are to be taken within the boot code.

All 68000 accesses to the dual port RAM are visible externally on the address and data pins so program execution in the 576 byte code space can be monitored.

After the boot process is completed by the user, it is suggested that the user issue the CP Reset command to the CP command register (CR) before reinitializing the SCCs. This will return the CP to its original state and eliminate any possible inconsistencies in the initialization process. The RISC cannot return to boot mode unless a system reset is executed with the PA7 pin asserted low. Toggling of the PA7 pin when the device is not in system reset is allowed, and in this mode the PA7 pin can be used in its alternate functions.

NOTE

The user may wish to disable the software watchdog timer (Timer 3) in the initial boot code if a long delay (i.e. more than 10 seconds) can occur between the initial boot download and the rest of the download process.

At the end of the Boot from SCC function, the following registers contain values that differ from their default reset values:

ICR = 0xc000
BAR = 0x0000
SCON1 = depends on mode.

The SCM1 register is reprogrammed to its reset value of 0x0

NOTE

During the boot from SCC procedure no external master should acquire the bus.

3.4 DMA CONTROL

The IMP includes seven on-chip DMA channels, six serial DMA (SDMA) channels for the three serial communications controllers (SCCs) and one IDMA.

3.4.1 68PM302 Differences

The $\overline{\text{DREQ}}$, $\overline{\text{DACK}}$, and $\overline{\text{DONE}}$ pins have been removed unless the PCMCIA interface is disabled. The user should not program the IDMA should for external request generation.

If the PCMCIA interface is disabled, then the $\overline{\text{DREQ}}$ pin is available on PA13. If the PCMCIA Interface is enabled, then the $\overline{\text{DREQ}}$ pin is available on PB8 (see 6.2 Summary of Pin Multiplexing for more details on pin muxing).

The external bus exceptions, $\overline{\text{BERR}}$ and retry, have been removed unless the PCMCIA interface is disabled. Only $\overline{\text{HALT}}$ or an internal $\overline{\text{BERR}}$ generated by the hardware watchdog timer or CS logic is supported.

The rest of the functionality remains the same as for the MC68302. For details on the bus operation, please refer to the *MC68302 User's Manual*.

3.4.1.1 Channel Mode Register (CMR). The CMR, a 16-bit register, is reset to $0000.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—	ECO	INTN	INTE	REQG		SAPI	DAPI	SSIZE		DSIZE		BT		RST	STR

Bit 15—Reserved for future use.

ECO—External Control Option (**Only supported if the PCMCIA is disabled**)

0 = If the request generation is programmed to be external in the REQG bits, the control signals ($\overline{\text{DACK}}$ and $\overline{\text{DONE}}$) are used in the source (read) portion of the transfer since the peripheral is the source.

1 = If the request generation is programmed to be external in the REQG bits, the control signals ($\overline{\text{DACK}}$ and $\overline{\text{DONE}}$) are used in the destination (write) portion of the transfer since the peripheral is the destination.

INTN—Interrupt Normal

0 = When the channel has completed an operand transfer without error conditions as indicated by $\overline{\text{DONE}}$, the channel does not generate an interrupt request to the IMP interrupt controller. The DONE bit remains set in the CSR.

1 = When the channel has completed an operand transfer without error conditions as indicated by $\overline{\text{DONE}}$, the channel generates an interrupt request to the IMP interrupt controller and sets DONE in the CSR.

NOTE

An interrupt will only be generated if the IDMA bit is set in the interrupt mask register (IMR).

INTE—Interrupt Error

0 = If a bus error occurs during an operand transfer either on the source read (BES) or the destination write (BED), the channel does not generate an interrupt to the IMP interrupt controller. The appropriate bit remains set in the CSR.

1 = If a bus error occurs during an operand transfer either on BES or BED, the channel generates an interrupt to the IMP interrupt controller and sets the appropriate bit (BES or BED) in the CSR.

NOTE

An interrupt will only be generated if the IDMA bit is set in the IMR.

REQG—Request Generation

00 = Internal request at limited rate (limited burst bandwidth) set by burst transfer (BT) bits

01 = Internal request at maximum rate (one burst)

10 = External request burst transfer mode-$\overline{\text{DREQ}}$ level sensitive

11 = External request cycle steal-$\overline{\text{DREQ}}$ edge sensitive

NOTE

The settings 10 and 11 will not work since the $\overline{\text{DREQ}}$ pin is not present.

SAPI—Source Address Pointer (SAP) Increment

0 = SAP is not incremented after each transfer.

1 = SAP is incremented by one or two after each transfer, according to the source size (SSIZE) bits and the starting address.

DAPI—Destination Address Pointer (DAP) Increment

0 = DAP is not incremented after each transfer.

1 = DAP is incremented by one or two after each transfer, according to the destination size (DSIZE) bits and the starting address.

SSIZE—Source Size

00 = Reserved
01 = Byte
10 = Word
11 = Reserved

DSIZE—Destination Size

00 = Reserved
01 = Byte
10 = Word
11 = Reserved

BT—Burst Transfer

00 = IDMA gets up to 75% of the bus bandwidth.
01 = IDMA gets up to 50% of the bus bandwidth.
10 = IDMA gets up to 25% of the bus bandwidth.
11 = IDMA gets up to 12.5% of the bus bandwidth.

RST—Software Reset

0 = Normal operation
1 = The channel aborts any external pending or running bus cycles and terminates channel operation. Setting RST clears all bits in the CSR and CMR.

STR—Start Operation

0 = Stop channel; clearing this bit will cause the IDMA to stop transferring data at the end of the current operand transfer. The IDMA internal state is not altered.
1 = Start channel; setting this bit will allow the IDMA to start (or continue if previously stopped) transferring data.

NOTE

STR is cleared automatically when the transfer is complete.

3.4.1.2 Source Address Pointer Register (SAPR)

31 24	23 0
RESERVED	SOURCE ADDRESS POINTER

The SAPR is a 32-bit register.

Note that A23-A20 must be initialized by the user. They are driven internally by the IDMA and can be used by the chip selects for address comparison.

3.4.1.3 Destination Address Pointer Register (DAPR). The DAPR is a 32-bit register.

31 24	23 0
RESERVED	DESTINATION ADDRESS POINTER

Note that A23-A20 must be initialized by the user. They are driven internally by the IDMA and can be used by the chip selects for address comparison.

3.4.1.4 Function Code Register (FCR). The FCR is an 8-bit register.

7	6 4	3	2 0
1	DFC	1	SFC

The function codes must e initialized by the user. The function code value programmed into the FCR is placed on pins FC2–FC0 during a bus cycle to further qualify the address bus value.

NOTE

> This register is undefined following power-on reset. The user should always initialize it and should not use the function code value "111" in this register.

3.4.1.5 Byte Count Register (BCR). This 16-bit register specifies the amount of data to be transferred by the IDMA; up to 64K bytes (BCR = 0) is permitted.

3.4.1.6 Channel Status Register (CSR). The CSR is an 8-bit register used to report events recognized by the IDMA controller. On recognition of an event, the IDMA sets its corresponding bit in the CSR (regardless of the INTE and INTN bits in the CMR).

7 4	3	2	1	0
RESERVED	DNS	BES	BED	DONE

Bits 7–4—These bits are reserved for future use.

DNS—Done Not Synchronized **(Only Valid if the PCMCIA is disabled)**

This bit is set if operand packing is performed between 16-bit memory and an 8-bit peripheral and the $\overline{\text{DONE}}$ signal is asserted as an input to the IDMA (i.e., by the peripheral) during the first access of the 8-bit peripheral.

BES—Bus Error Source

This bit indicates that the IDMA channel terminated with an error during the read cycle.

BED—Bus Error Destination

This bit indicates that the IDMA channel terminated with an error during the write cycle.

DONE—Normal Channel Transfer Done

This bit indicates that the IDMA channel has terminated normally.

3.4.2 Interface Signals

When the PCMCIA interface is disabled, the IDMA channel has three dedicated control signals: DMA request ($\overline{\text{DREQ}}$), DMA acknowledge ($\overline{\text{DACK}}$), and end of IDMA transfer ($\overline{\text{DONE}}$). The IDMA's use of the bus arbitration signals is described in the *MC68302 Users' Manual*. IDMA Operational Description

NOTE

For details on the IDMA programming and bus operation, please refer to Section 3 of the *MC68302 Users' Manual*

3.5 INTERRUPT CONTROLLER

The IMP interrupt controller accepts and prioritizes both internal and external interrupt requests and generates a vector number during the CPU interrupt acknowledge cycle.

3.5.1 Interrupt Controller Key Differences from the MC68302

Since the function code pins are not connected externally unless the PCMCIA interface is disabled (or the PGA Package is used), the 68PM302 should be programmed to Dedicated Mode and to internally generate the vectors for Levels 1, 6, and 7. An external device will not easily be able to decode an IACK cycle and provide an vector back to the 68PM302.

In Disable CPU mode with the PCMCIA interface enabled, the $\overline{\text{IRQ1}}$, $\overline{\text{IRQ6}}$, and $\overline{\text{IRQ7}}$ become the $\overline{\text{BR}}$, $\overline{\text{BGACK}}$, and $\overline{\text{BG}}$ signals. With the core disabled, the MC68PM302 will not be able to decode an external CPU's interrupt acknowledge cycle. The user must poll the Interrupt Pending Register (IPR) during interrupt handling to determine which peripheral caused the interrupt. If the PCMCIA interface is disabled, then the MC68PM302 can use the function codes to decode the interrupt acknowledge cycle and provide a vector back to the host CPU.

3.5.1.1 Interrupt Controller Overview. In addition to the MC68302 internal sources, the interrupt controller receives interrupts from the PCMCIA Controller and from the Fast Wake-Up Ring Oscillator (see 2.4.4.3 Fast Wake-Up). These interrupts are additional sources to $\overline{\text{IRQ7}}$, $\overline{\text{IRQ6}}$, and $\overline{\text{IRQ1}}$. (programmable in PCMR, DCOR and RINGOCR registers). With PCMCIA Interrupts it is possible to work in Normal or Dedicated mode for the three external interrupt requests pins. The chip ORs all sources at the same level and asserts the interrupt to the 68000 core. Interrupts at the same level (1,6 or 7) are not prioritized.

3.5.2 Interrupt Controller Programming Model

The user communicates with the interrupt controller using four registers. The global interrupt mode register (GIMR) defines the interrupt controller's operational mode. The interrupt pending register (IPR) indicates which INRQ interrupt sources require interrupt service. The interrupt mask register (IMR) allows the user to prevent any of the INRQ interrupt sources from generating an interrupt request. The interrupt in-service register (ISR) provides a capability for nesting INRQ interrupt requests.

3.5.2.1 Global Interrupt Mode Register (GIMR). The user normally writes the GIMR soon after a total system reset. The GIMR is initially $0000 and is reset only upon a total system reset.

15	14	13	12	11	10	9	8	7	5	4	0
MOD	IV7	IV6	IV1	—	ET7	ET6	ET1	V7–V5		RESERVED	

MOD—Mode

0 = Normal operational mode. Interrupt request lines are configured as $\overline{\text{IPL2}}$–$\overline{\text{IPL0}}$.
1 = Dedicated operational mode. Interrupt request lines are configured as $\overline{\text{IRQ7}}$, $\overline{\text{IRQ6}}$, and $\overline{\text{IRQ1}}$.

IV7—Level 7 Interrupt Vector

0 = Internal vector.
1 = External vector

IV6—Level 6 Interrupt Vector

0 = Internal vector.
1 = External vector.

IV1—Level 1 Interrupt Vector

0 = Internal vector.
1 = External vector.

ET7—$\overline{\text{IRQ7}}$ Edge-/Level-Triggered

0 = Level-triggered. An interrupt is made pending when $\overline{\text{IRQ7}}$ is low.
1 = Edge-triggered. An interrupt is made pending when $\overline{\text{IRQ7}}$ changes from one to zero (falling edge).

ET6—IRQ6 Edge-/Level-Triggered

0 = Level-triggered. An interrupt is made pending when $\overline{\text{IRQ6}}$ is low.
1 = Edge-triggered. An interrupt is made pending when $\overline{\text{IRQ6}}$ changes from one to zero (falling edge).

ET1—$\overline{\text{IRQ1}}$ Edge-/Level-Triggered

0 = Level-triggered. An interrupt is made pending when $\overline{\text{IRQ1}}$ is low.
1 = Edge-triggered. An interrupt is made pending when $\overline{\text{IRQ1}}$ changes from one to zero (falling edge).

V7–V5—Interrupt Vector Bits 7–5

These three bits are concatenated with five bits provided by the interrupt controller, which indicate the specific interrupt source, to form an 8-bit interrupt vector number. If these bits are not written, the vector $0F is provided.

NOTE:

These three bits should be greater than or equal to ‘010’ in order to put the interrupt vector in the area of the exception vector table for user vectors.

Bits 11 and 4–0—Reserved for future use.

3.5.2.2 Interrupt Pending Register (IPR). Each bit in the 16-bit IPR corresponds to an INRQ interrupt source. When an INRQ interrupt is received, the interrupt controller sets the corresponding bit in the IPR.

15	14	13	12	11	10	9	8
PB11	PB10	SCC1	SDMA	IDMA	SCC2	TIMER1	SCC3

7	6	5	4	3	2	1	0
PB9	TIMER2	SCP	TIMER3	SMC1	SMC2	PB8	ERR

3.5.2.3 Interrupt Mask Register (IMR). Each bit in the 16-bit IMR corresponds to an INRQ interrupt source. The user masks an interrupt source by clearing the corresponding bit in the IMR.

15	14	13	12	11	10	9	8
PB11	PB10	SCC1	SDMA	IDMA	SCC2	TIMER1	SCC3

7	6	5	4	3	2	1	0
PB9	TIMER2	SCP	TIMER3	SMC1	SMC2	PB8	—

3.5.2.4 Interrupt In-Service Register (ISR) . Each bit in the 16-bit ISR corresponds to an INRQ interrupt source. In a vectored interrupt environment, the interrupt controller sets the ISR bit when the vector number corresponding to the INRQ interrupt source is passed to the core during an interrupt acknowledge cycle. The user's interrupt service routine should clear this bit during the servicing of the interrupt.

15	14	13	12	11	10	9	8
PB11	PB10	SCC1	SDMA	IDMA	SCC2	TIMER1	SCC3

7	6	5	4	3	2	1	0
PB9	TIMER2	SCP	TIMER3	SMC1	SMC2	PB8	0

3.6 PARALLEL I/O PORTS

The IMP supports three general-purpose I/O ports, port A, port B, and port D whose pins can be general-purpose I/O pins or dedicated peripheral interface pins. Some port B pins are always maintained as four general-purpose I/O pins, each with interrupt capability.

3.6.1 Port A

Each of the 16 port A pins are independently configured as a general-purpose I/O pin if the corresponding port A control register (PACNT) bit is cleared.

NOTE

> PA7-0 are only available if the PCMCIA interface is disabled or the PCMCIA data bus is only 8 bits wide. PA15-13 are only available when the PCMCIA interface is disabled.

Port A pins are configured as dedicated on-chip peripheral pins if the corresponding PACNT bit is set. When acting as a general-purpose I/O pin, the signal direction for that pin is determined by the corresponding control bit in the port A data direction register (PADDR). (Note that these port pins do not have internal pullup resistors for non-PCMCIA mode). If a port A pin is selected as a general-purpose I/O pin, it may be accessed through the port A data register (PADAT).

Table 3-2. Port A Pin Functions

PACNT Bit = 1 Pin Function	PACNT Bit = 0 Pin Function	Input to SCC2/SCC3/IDMA
RXD2	PAO	GND
TXD2	PA1	—
RCLK2	PA2	GND
TCLK2	PA3	RCLK2 #
$\overline{\text{CTS2}}$	PA4	GND
$\overline{\text{RTS2}}$	PA5	—
$\overline{\text{CD2}}$	PA6	GND
SDS2/BRG2	PA7	—
RXD3	PA8	GND
TXD3	PA9	—
RCLK3	PA10	GND
TCLK3	PA11	RCLK3 #
BRG3	PA12	—
$\overline{\text{DREQ}}$	PA13	GND
$\overline{\text{DACK}}$	PA14	—
$\overline{\text{DONE}}$	PA15	V_{DD}

# Allows a single external clock source on the RCLK pin to clock both the SCC receiver and transmitter.

3.6.2 Port B

Port B has 12 pins. PB7–PB0 may be configured as general-purpose I/O pins or as dedicated peripheral interface pins; whereas, PB11–PB8 are always maintained as four general-purpose pins, each with interrupt capability.

NOTE

PB2-0, and PB4 are only available when the PCMCIA interface is disabled.

3.6.2.1 PB7–PB0. Each port B pin may be configured as a general-purpose I/O pin or as a dedicated peripheral interface pin. PB7–PB0 functions exactly like PA15–PA0, except that PB7–PB0 is controlled by the port B control register (PBCNT), the port B data direction register (PBDDR), and the port B data register (PBDAT), and PB7 is configured as an open-drain output ($\overline{\text{WDOG}}$) upon total system reset.

Table 3-3 shows the dedicated function of each pin. The third column shows the input to the peripheral when the pin is used as a general-purpose I/O pin.

Table 3-3. Port B Pin Functions

PBCNT Bit = 1 Pin Function	PBCNT Bit = 0 Pin Function	Input to Interrupt Control and Timers
IACK7	PB0	—
IACK6	PB1	—
IACK1	PB2	—
TIN1	PB3	GND
TOUT1	PB4	—
TIN2	PB5	GND
TOUT2	PB6	—
WDOG	PB7	—

3.6.2.2 PB11–PB8. PB11–PB8 are four general-purpose I/O pins continuously available as general-purpose I/O pins and, therefore, are not referenced in the PBCNT. PB8 operates like PB11–PB9 except that it can also be used as the DRAM refresh controller request pin, as selected in the system control register (SCR).

NOTE

> If the PIT is enabled, then the PB8 pin will not generate an interrupt, since the PIT uses the PB8 interrupt in the IPR, IMR, and ISR.

3.6.3 Port D

When the PCMCIA interface is disabled (PC_EN tied to GND), 7 general purpose **INPUT ONLY** pins are available as port D.These pins also have optional internal pull-up resistors (see Pullup Control Register (PUCR) on page 20)**.**

Port D may be accessed through the Port D data register (PDDAT). When PDDAT is read, the state of the input pins is read.

3.6.4 Port Registers

The I/O port consists of three memory-mapped read-write 16-bit registers for port A and three memory-mapped read-write 16-bit registers for port B. Refer to Figure 3-2 for the I/O port registers. The reserved bits are read as zeros.

Port A Control Register (PACNT)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CA	CA	CA	CA	CA	CA	CA	CA	CA	CA	CA	CA	CA	CA	CA	CA

0 = I/O 1 = Peripheral

Port A Data Direction Register (PADDR)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DA	DA	DA	DA	DA	DA	DA	DA	DA	DA	DA	DA	DA	DA	DA	DA

0 = Input 1 = Output

Port A Data Register (PADAT)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PA	PA	PA	PA	PA	PA	PA	PA	PA	PA	PA	PA	PA	PA	PA	PA

Port B Control Register (PBCNT)

15							8	7	6	5	4	3	2	1	0
RESERVED								CB	CB	CB	CB	CB	CB	CB	CB

0 = I/O   1 = Peripheral

Port B Data Direction Register (PBDDR)

15			12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				DB	DB	DB	DB	DB	DB	DB	DB	DB	DB	DB	DB

0 = Input   1 = Output

Port B Data Register (PBDAT)

15			12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				PB	PB	PB	PB	PB	PB	PB	PB	PB	PB	PB	PB

Port D Data Register (PDDAT)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	PD	PD	PD	PD	PD	PD	PD	0	0	0	0	0	0	0	0

Figure 3-2. Parallel I/O Port Registers

3.7 TIMERS

The MC68PM302 includes four timer units: two identical general-purpose timers, a software watchdog timer, and a periodic interrupt timer (PIT).

Each general-purpose timer consists of a timer mode register (TMR), a timer capture register (TCR), a timer counter (TCN), a timer reference register (TRR), and a timer event register (TER). The TMR contains the prescaler value programmed by the user. The software watchdog timer, which has a watchdog reference register (WRR) and a watchdog counter (WCN), uses a fixed prescaler value.

3.7.1 General Purpose Timers Programming Model

3.7.1.1 Timer Mode Register (TMR1, TMR2). TMR1 and TMR2 are identical 16-bit registers. TMR1 and TMR2, which are memory-mapped read-write registers to the user, are cleared by reset.

15							8	7	6	5	4	3	2	1	0
PRESCALER VALUE (PS)								CE		OM	ORI	FRR	ICLK		RST

RST—Reset Timer

0 = Reset timer (software reset), includes clearing the TMR, TRR, and TCN.
1 = Enable timer

ICLK—Input Clock Source for the Timer

00 = Stop count
01 = Master clock
10 = Master clock divided by 16
11 = Corresponding TIN pin, TIN1 or TIN2 (falling edge)

FRR—Free Run/Restart

0 = Free run—timer count continues to increment after the reference value is reached.
1 = Restart—timer count is reset immediately after the reference value is reached.

ORI—Output Reference Interrupt Enable

0 = Disable interrupt for reference reached
1 = Enable interrupt upon reaching the reference value

OM—Output Mode

0 = Active-low pulse for one CLKO clock cycle (60 ns at 16.67 MHz)
1 = Toggle output

CE—Capture Edge and Enable Interrupt

00 = Capture function is disabled
01 = Capture on rising edge only and enable interrupt on capture event
10 = Capture on falling edge only and enable interrupt on capture event
11 = Capture on any edge and enable interrupt on capture event

PS—Prescaler Value

The prescaler is programmed to divide the clock input by values from 1 to 256. The value 00000000 divides the clock by 1; the value 11111111 divides the clock by 256.

3.7.1.2 Timer Reference Registers (TRR1, TRR2). Each TRR is a 16-bit register containing the reference value for the timeout. TRR1 and TRR2 are memory-mapped read-write registers.

3.7.1.3 Timer Capture Registers (TCR1, TCR2). Each TCR is a 16-bit register used to latch the value of the counter during a capture operation when an edge occurs on the respective TIN1 or TIN2 pin. TCR1 and TCR2 appear as memory-mapped read-only registers to the user.

3.7.1.4 Timer Counter (TCN1, TCN2). TCN1 and TCN2 are 16-bit up-counters. Each is memory-mapped and can be read and written by the user. A read cycle to TCN1 and TCN2 yields the current value of the timer and does not affect the counting operation.

3.7.1.5 Timer Event Registers (TER1, TER2). Each TER is an 8-bit register used to report events recognized by any of the timers. On recognition of an event, the timer will set the appropriate bit in the TER, regardless of the corresponding interrupt enable bits (ORI and CE) in the TMR. TER1 and TER2, which appear to the user as memory-mapped registers, may be read at any time. A bit is cleared by writing a one to that bit (writing a zero does not affect a bit's value).

7 ... 2	1	0
RESERVED	REF	CAP

CAP—Capture Event

The counter value has been latched into the TCR. The CE bits in the TMR are used to enable the interrupt request caused by this event.

REF—Output Reference Event

The counter has reached the TRR value. The ORI bit in the TMR is used to enable the interrupt request caused by this event.

Bits 7–2—Reserved for future use.

3.7.2 Timer 3 - Software Watchdog Timer

A watchdog timer is used to protect against system failures by providing a means to escape from unexpected input conditions, external events, or programming errors. Timer 3 may be used for this purpose. Once started, the watchdog timer must be cleared by software on a regular basis so that it never reaches its timeout value. Upon reaching the timeout value, the assumption may be made that a system failure has occurred, and steps can be taken to recover or reset the system. No changes have been made to the software watchdog timer. Please refer to the *MC68302 Users' Manual* for more information.

3.7.2.1 Software Watchdog Reference Register (WRR). WRR is a 16-bit register containing the reference value for the timeout. The EN bit of the register enables the timer. WRR appears as a memory-mapped read-write register to the user.

15 ... 1	0
REFERENCE VALUE	EN

3.7.2.2 Software Watchdog Counter (WCN). WCN, a 16-bit up-counter, appears as a memory-mapped register and may be read at any time. Clearing EN in WRR causes the counter to be reset and disables the count operation. A read cycle to WCN causes the current value of the timer to be read. A write cycle to WCN causes the counter and prescaler to be reset. A write cycle should be executed on a regular basis so that the watchdog timer is never allowed to reach the reference value during normal program operation.

3.7.3 Periodic Interrupt Timer (PIT)

The MC68PM302 provides a timer to generate periodic interrupts for use with a real-time operating system or the application software. The periodic interrupt time period can vary from 122 µs to 128 s (assuming a 32.768-kHz crystal is used to generate the general system clock). This function can be disabled.

3.7.3.1 Overview. The periodic interrupt timer consists of an 11-bit modulus counter that is loaded with the value contained in the PITR. The modulus counter is clocked by the CLKIN signal derived from the IMP EXTAL pin. See Figure 2-3.

The clock source is divided by four before driving the modulus counter (PITCLK). When the modulus counter value reaches zero, an interrupt request signal is generated to the IMP interrupt controller.

The value of bits 11–1 in the PITR is then loaded again into the modulus counter, and the counting process starts over. A new value can be written to the PITR only when the PIT is disabled.

The PIT interrupt replaces the IMP PB8 interrupt and is mapped to the PB8 interrupt priority level 4. The PIT interrupt can be masked using the corresponding bit in the IMR.

NOTE

When the PIT is enabled, PB8 can still be used as parallel I/O, $\overline{\text{DREQ}}$, $\overline{\text{PC\_DREQ}}$ pin or as DRAM refresh controller request pin, but PB8 will not be capable of generating interrupts.

3.7.3.2 Periodic Timer Period Calculation. The period of the periodic timer can be calculated using the following equation:

$$\text{periodic interrupt timer period} = \frac{\text{PITR count value+1}}{\cfrac{((EXTAL)/(1 or 512))}{(4)}}$$

Solving the equation using a crystal frequency of 32.768 kHz with the prescaler disabled gives:

$$\text{periodic interrupt timer period} = \frac{\text{PITR count value+1}}{\cfrac{32768/1}{2^2}}$$

$$\text{periodic interrupt timer period} = \frac{\text{PITR count value}}{8192}$$

This gives a range from 122 µs, with a PITR value of \$0, to 250 ms, with a PITR value of \$7FF (assuming a 32.768Khz at the EXTAL pin.

Solving the equation with the prescaler enabled (PTP=1) gives the following values:

$$\text{periodic interrupt timer period} = \frac{\text{PITR count value}}{\frac{(32768/512)}{2^2}}$$

$$\text{periodic interrupt timer period} = \frac{\text{PITR count value}}{16}$$

This gives a range from 62.5 ms, with a PITR value of $0 to 128 s, with a PITR value of $7FF.

For a fast calculation of periodic timer period using a 32.768-kHz crystal, the following equations can be used:

With prescaler disabled:

programmable interrupt timer period = PITR (122 µs)

With prescaler enabled:

programmable interrupt timer period = PITR (62.5 ms)

3.7.3.3 Using the Periodic Timer As a Real-Time Clock. The periodic interrupt timer can be used as a real-time clock interrupt by setting it up to generate an interrupt with a one-second period. When using a 32.768-kHz crystal, the PITR should be loaded with a value of $0F with the prescaler enabled to generate interrupts at a one-second rate. The interrupt is generated, in this case, at a precise 1 second rate, even if the interrupt is not serviced immediately. A true real time clock is obtained if the current interrupt is serviced completely before the next one occurs.

3.7.3.4 Periodic Interrupt Timer Register (PITR). The PITR contains control for prescaling the periodic timer as well as the count value for the periodic timer. This register can be read or written only during normal operational mode. Bits 14–13 are not implemented and always return a zero when read. A write does not affect these bits.

PITR $0F0

	15	14	13	12	11	10	9	8
	PTEN	0	0	PTP	PITR10	PITR9	PITR8	PITR7
RESET	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
	PITR6	PITR5	PITR4	PITR3	PITR2	PITR1	PITR0	RES
RESET	0	0	0	0	0	0	0	0

Read/Write

PTEN—Periodic Timer Enable

This bit contains the enable control for the periodic timer.

0 = Periodic timer is disabled
1 = Periodic timer is enabled

PTP—Periodic Timer Prescaler Control

This bit contains the prescaler control for the periodic timer.

0 = Periodic timer clock is not prescaled
1 = Periodic timer clock is prescaled by a value of 512

PITR10–0—Periodic Interrupt Timer Register Bits

These bits of the PITR contain the remaining bits of the PITR count value for the periodic timer. **These bits may be written only when the PIT is disabled (PTEN=0) to modify the PIT count value.**

NOTE

If the PIT is enabled with the PTP bit is set, the first interrupt can be up to 512 clocks early, depending on the prescaler counter value when the PIT is enabled.

3.8 EXTERNAL CHIP-SELECT SIGNALS AND WAIT-STATE LOGIC

The IMP provides a set of four programmable chip-select signals. Each chip-select signal has an identical internal structure. For each memory area, the user may also define an internally generated cycle termination signal ($\overline{\text{DTACK}}$). This feature eliminates board space that would be necessary for cycle termination logic.

The chip-select logic is active for memory cycles generated by internal bus masters (M68000 core, IDMA, SDMA, DRAM refresh, PCMCIA controller) or external bus masters **(A23-A20 are driven to zero internally and FC2-0 are driven to 5)**. These signals are driven externally on the falling edge of $\overline{\text{AS}}$ and are valid shortly after $\overline{\text{AS}}$ goes low.

NOTE

For more information on the operation of the Chip Selects, please refer to Section 3 of the *MC68302 Users' Manual*.

Also, note that A23-20 and FC2-0 are not driven externally unless the PCMCIA interface is disabled; however, these signals are still used internally by the chip selects for comparison.

In Disable CPU mode or for external bus masters, A23-A20 are internally driven to zero, so the user must program the corresponding bits in the Chip Select registers to zero, or mask them off.

Also FC2-0 are driven to 5, so we suggest that the function code comparison be turned off.

3.8.1 Chip-Select Registers

Each of the four chip-select units has two registers that define its specific operation. These registers are a 16-bit base register (BR) and a 16-bit option register (OR) (e.g., BR0 and OR0). The BR should normally be programmed after the OR since the BR contains the chip-select enable bit.

3.8.1.1 Base Register (BR3–BR0). These 16-bit registers consist of a base address field, a read-write bit, and a function code field.

15 13	12 2	1	0
FC2 –FC0	BASE ADDRESS (A23–A13)	RW	EN

FC2–FC0 —Function Code Field

This field is contained in bits 15–13 of each BR. These bits are used to set the address space function code. Because of the priority mechanism and the EN bit, only the $\overline{\text{CS0}}$ line is active after a system reset.

Bits 12–2—Base Address

These bits are used to set the starting address of a particular address space.

RW—Read/Write

0 = The chip-select line is asserted for read operations only.
1 = The chip-select line is asserted for write operations only.

EN—Enable

0 = The chip-select line is disabled.
1 = The chip-select line is enabled.

After system reset, only $\overline{\text{CS0}}$ is enabled; $\overline{\text{CS3}}$–$\overline{\text{CS1}}$ are disabled. In disable CPU mode, $\overline{\text{CS3}}$–$\overline{\text{CS0}}$ are disabled at system reset. The chip select does not require disabling before changing its parameters.

3.8.1.2 Option Registers (OR3–OR0). These four 16-bit registers consist of a base address mask field, a read/write mask bit, a compare function code bit, and a $\overline{\text{DTACK}}$ generation field.

15 13	12 2	1	0
DTACK	BASE ADDRESS MASK (M23–M13)	MRW	CFC

Bits 15–12—DTACK Field

These bits are used to determine whether $\overline{\text{DTACK}}$ is generated internally with a programmable number of wait states or externally by the peripheral.

Table 3-4. DTACK Field Encoding

Bits			Description
15	14	13	
0	0	0	No Wait State
0	0	1	1 Wait State
0	1	0	2 Wait States
0	1	1	3 Wait States
1	0	0	4 Wait States
1	0	1	5 Wait States
1	1	0	6 Wait States
1	1	1	External DTACK

Bits 12–2—Base Address Mask

These bits are used to set the block size of a particular chip-select line. The address compare logic uses only the address bits that are not masked (i.e., mask bit set to one) to detect an address match.

0 = The address bit in the corresponding BR is masked.
1 = The address bit in the corresponding BR is not masked.

MRW—Mask Read/Write

Should be disabled in enable CPU - enable PCMCIA mode if an external master uses the CS logic for its memory space.

0 = The RW bit in the BR is masked.
1 = The RW bit in the BR is not masked.

CFC—Compare Function Code

Should be disabled in PCMCIA mode if an external master uses the CS logic for its memory space.

0 = The FC bits in the BR are ignored.
1 = The FC bits on the BR are compared.

3.8.1.3 PCMCIA Protection Register (PPR). This register controls the PCMCIA protection mechanism that blocks the chip select mechanism for external PCMCIA accesses made to the block of memory covered by the protected chip select. It also includes the Protect Internal Registers (PIR) bit that controls the protection mechanism to block PCMCIA accesses to the internal registers and dual port RAM. This register is located at BAR+82C. This register is cleared at reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	BSW	BSWEN	PIR	PCS3	PCS2	PCS1	PCS0

PCS(3-0)—Protect Chip Select

Each of these bits when set will disable the corresponding chip select for PCMCIA accesses. When a PCMCIA controller access is made to a protected memory space, the chip select line will not assert and the DTACK line will also not be asserted. This will result in a 68000 bus error and an Interrupt to the PCMCIA (if the BERRIE bit is set in the PCMR register).

PIR—Protect Internal Registers

This bit when set will block PCMCIA accesses to internal registers and the internal dual port ram. When a PCMCIA access is made to an internal register or memory location while the PIR bit is set, the register will not be addressed and the DTACK line will not be asserted. This will result in a 68000 bus error and an Interrupt to the PCMCIA (if the BERRIE bit is set in the PCMR register).

BSWEN - Bus Width Switch Enable

When this bit is toggled from a zero to a one, the bus width switch mechanism is enabled. From the point this bit is toggled, the bus width is determined by the BSW bit of this register. If another bus width switch is necessary, this bit must be toggled back to zero and then one again. This bit is zero following a power up reset.

Setting this bit enables a hardware state machine that arbitrates the internal bus away from the 302 core, changes the BUSW pin internally and then gives the bus back to the 302 core).

BSW - Bus Width

This bit determines the bus width after the bus width switch is performed.

0 - Data Bus width is 8 bits
1 - Data Bus width is 16 bits.

3.8.2 Disable CPU Logic (M68000)

The IMP can be configured to operate solely as a peripheral to an external processor. In this mode, the on-chip M68000 CPU should be disabled by strapping DISCPU high during system reset ($\overline{\text{RESET}}$ and $\overline{\text{HALT}}$ asserted simultaneously). The internal accesses to the IMP peripherals and memory may be asynchronous or synchronous. During synchronous reads, one wait state may be used if required (EMWS bit set). The following pins change their functionality in this mode:

1. THE $\overline{\text{IPL0}}$ pin becomes $\overline{\text{BR}}$ and is an output from the IDMA and SDMA to the external M68000 bus.
2. The $\overline{\text{IPL2}}$ pin becomes $\overline{\text{BG}}$ and is an input to the IDMA and SDMA from the external M68000 bus. When BG is sampled as low by the IMP, it waits for $\overline{\text{AS}}$, $\overline{\text{HALT}}$, and $\overline{\text{BGACK}}$ to be negated, and then asserts $\overline{\text{BGACK}}$ and performs one or more bus cycles.
3. The $\overline{\text{IPL1}}$ pin becomes $\overline{\text{BGACK}}$ and is an output from the IDMA and SDMA to indicate bus ownership.
4. The $\overline{\text{IPL2-0}}$ lines are no longer encoded interrupt lines.The interrupt controller will output the 68PM302’s interrupt request on $\overline{\text{IOUT2}}$. $\overline{\text{CS0}}$, which is multiplexed with $\overline{\text{IOUT2}}$ is not available in this mode.
5. The $\overline{\text{WEH}}$ and $\overline{\text{WEL}}$ signals become UDS and LDS respectively.
6. The $\overline{\text{OE}}$ becomes R/$\overline{\text{W}}$.

NOTE

> If the PCMCIA interface is disabled, then the $\overline{\mathrm{IPL2\text{-}0}}$, $\overline{\mathrm{WEH}}$, $\overline{\mathrm{WEL}}$, and $\overline{\mathrm{OE}}$ signals are available along with $\overline{\mathrm{BR}}$, $\overline{\mathrm{BG}}$, $\overline{\mathrm{BGACK}}$, $\overline{\mathrm{UDS}}$, $\overline{\mathrm{LDS}}$, and R/$\overline{\mathrm{W}}$.

DISCPU should remain continuously high during disable CPU mode operation. Although the $\overline{\mathrm{CS0}}$ pin is not available as an output from the device in disable CPU mode, it may be enabled to provide $\overline{\mathrm{DTACK}}$ generation. In disable CPU mode, BR0 is initially $C000.

In Disable CPU mode, accesses by an external master to the IMP RAM and registers may be asynchronous or synchronous to the IMP clock. See the SAM and EMWS bits in the SCR for details.

In Disable CPU Mode, the PM302 can interrupt an external CPU with the $\overline{\mathrm{CS0}}$/IOUT2 signal only (IOUT1 and IOUT0 have been deleted). IOUT2 is asserted for interrupts of Level 4, 6, 7 which are generated by internal Blocks.

VGE—Vector Generation Enable

This bit should be cleared unless the PCMCIA interface is disabled. Otherwise the MC68PM302 cannot decode an interrupt acknowledge cycle without the function code pins.

0 = In disable CPU mode, the IMP will not output interrupt vectors during interrupt acknowledge cycles.

1 = In disable CPU mode, the IMP will output interrupt vectors for internal level 4 interrupts (and for levels 1, 6, and/or 7 as enabled in the interrupt controller) during interrupt acknowledge cycles.

3.8.3 Bus Arbitration Logic

Both internal and external bus arbitration are discussed in the following paragraphs.

3.8.3.1 Internal Bus Arbitration.

In disable CPU mode, or when the PCMCIA interface is disabled, the IMP bus arbiter supports three bus request sources in the following standard priority:

1. External bus master ($\overline{\mathrm{BR}}$ pin if the PCMCIA interface is disabled or in Disable CPU mode)
2. SDMA for the SCCs (six channels)
3. IDMA (one channel)
4. PCMCIA Controller

In the enable CPU mode if the PCMCIA Interface is enabled, External Master can acquire the bus by asserting $\overline{\mathrm{HALT}}$ rather than using the normal arbitration mechanism.

In the disable CPU mode, the PM302 makes requests for the bus rather than granting the bus. In such a system, the IMP functions as an external master, and the external processor (e.g., an MC68030) does not assert $\overline{\text{BGACK}}$ as it accesses the IMP's on-chip RAM and registers.

3.8.3.2 External Bus Arbitration Using $\overline{\text{HALT}}$.

When the CPU is enabled and the PCMCIA interface is enabled, an external bus master may gain ownership of the M68000 bus by asserting the $\overline{\text{HALT}}$ signal. In order to enable external bus arbitration in this mode, the EMEN bit in the SCR should be set. Assertion of the $\overline{\text{HALT}}$ signal will cause the PM302 bus master (M68000 core, SDMA, IDMA or PCMCIA) to stop at the completion of the current bus cycle; the external bus master must wait until $\overline{\text{AS}}$ is negated, plus 2 additional system clocks, before accessing the bus (to allow the PM302 to threestate all of the bus signals). After gaining ownership, the external master can not access the internal IMP registers or RAM. Chip select logic and system control functions, such as the hardware watchdog, continue to operate.

When an external master desires to gain ownership, the following bus arbitration protocol should be used:

1. Assert $\overline{\text{HALT}}$.
2. If bit FAST=0 in PCMR wait two system clocks. If bit FAST=1 in PCMR wait three system clocks.
3. If $\overline{\text{AS}}$ is negated go to step 5.
4. Wait for $\overline{\text{AS}}$ negation. Wait two additional system clocks.
5. Execute Access (now the bus is guaranteed to be threestated).
6. When done, threestate bus and Negate $\overline{\text{HALT}}$.

NOTE

The RMCST bit in the SCR should be zero for this arbitration procedure to work correctly.

Also, the external master cannot access the internal address space of the 68PM302.

NOTE

$\overline{\text{WEH}}$, $\overline{\text{WEL}}$ and $\overline{\text{OE}}$ are threestated when HALT is asserted by external master.

3.9 DYNAMIC RAM REFRESH CONTROLLER

The communications processor (CP) main (RISC) controller may be configured to handle the dynamic RAM (DRAM) refresh task without any intervention from the M68000 core. Use of this feature requires a timer or SCC baud rate generator (either from the IMP or externally), the I/O pin PB8, and two transmit buffer descriptors from SCC2 (Tx BD6 and Tx BD7).

No changes have been made to the DRAM controller. For more information, please refer to the *MC68302 Users' Manual*.

SECTION 4 COMMUNICATIONS PROCESSOR (CP)

The CP includes the following modules:

- Main Controller (RISC Processor)
- Six Serial Direct Memory Access (SDMA) Channels
- A Command Set Register
- Serial Channels Physical Interface Including:

  —Motorola Interchip Digital Link (IDL)
  —General Circuit Interface (GCI), Also Known as IOM-2
  —Pulse Code Modulation (PCM) Highway Interface
  —Nonmultiplexed Serial Interface (NMSI) Implementing Standard Modem Signals
- Three Independent Full Duplex Serial Communication Controllers (SCCs) Supporting the Following Protocols:

  —High-Level/Synchronous Data Link Control (HDLC/SDLC)
  —Universal Asynchronous Receiver Transmitter (UART)
  —Binary Synchronous Communication (BISYNC)
  —Transparent Modes
  —Uart 16550 Hardware and Software Emulation
- Serial Communication Port (SCP) for Synchronous Communication
- Two Serial Management Controllers (SMCs) to Support the IDL and GCI Management Channels

4.1 MC68PM302 KEY DIFFERENCES FROM THE MC68302

- SCC2 Is Only Available when the PCMCIA Is Used in 8-Bit Data Operation and the 16550 Channel Is Disabled.
- The DDCMP and V.110 Protocols Were Removed.
- The Autobaud Function Was Added for Detecting the Baud Rate of the Incoming Asynchronous Bit Stream.

This section only presents a description of features and registers that have changed or been added. Features that have not changed such as UART, HDLC, BISYNC Transparent, the SMCs, and the SCP will not be discussed. For more information on any function not discussed in this section, please refer to the *MC68302 User’s Manual*.

This section assumes that the user is familiar with the different protocols. For more information on a specific protocol implementation, please refer to the *MC68302 User’s Manua*l

4.2 SERIAL CHANNELS PHYSICAL INTERFACE

The serial channels physical interface joins the physical layer serial lines to the three SCCs and the two SMCs. (The separate three-wire SCP interface is described in Serial Communication Port (SCP) on page 48.)

The IMP supports five different external physical interfaces from the SCCs:

1. NMSI—Nonmultiplexed Serial Interface
2. PCM—Pulse Code Modulation Highway
3. IDL—Interchip Digital Link
4. GCI—General Circuit Interface
5. UART 16550 physical register set through the PCMCIA interface

4.2.1 Serial Interface Registers

There are two serial interface registers: SIMODE and SIMASK. The SIMODE register is a 16-bit register used to define the serial interface operation modes. The SIMASK register is a 16-bit register used to determine which bits are active in the B1 and B2 channels of ISDN. For more information on these two registers, please refer to the *MC68302 Users' Manual*.

4.3 SERIAL COMMUNICATION CONTROLLERS (SCCS)

The IMP contains three independent SCCs, each of which can implement different protocols. This configuration provides the user with options for controlling up to three independent full-duplex lines implementing bridges or gateway functions or multiplexing up to three SCCs onto the same physical layer interface to implement a 2B + D ISDN basic rate channel or three channels of a PCM highway. Each protocol-type implementation uses identical buffer structures to simplify programming.

4.3.1 SCC Configuration Register (SCON)

Each SCC controller has a configuration register that controls its operation and selects its clock source and baud rate. This register has not been changed from the MC68302.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WOMS	EXTC	TCS	RCS	CD10	CD9	CD8	CD7	CD6	CD5	CD4	CD3	CD2	CD1	CD0	DIV4

4.3.1.1 DIVIDE BY 2 INPUT BLOCKS (NEW FEATURE). The SCC Baud Rate Generators have 2 divide by 2 blocks added to them. With the divide by 2 blocks enabled, the VCO Output from the PLL and the TIN1 input clock can be divided by 2 before they are used by the BRG to generate the serial clocks. The divide by two blocks can be enabled by setting the BCD bit in the IOMCR register if the BRG clock source is derived from the IMP system clock, or by setting the BRGDIV bit in the DISC register if the BRG clock source is derived from the TIN pin.

4.3.2 Baud Rate Generator Pins Register–BRGP

This 8 bit register controls: (1) enabling the divide by 2 prescaler for the baud rate generator from the TIN1 pin, and (2) options for three stating the BRG1, TCLK1, and RCLK1 pins.

BRGP Base+$8EE

15	14	13	12	11	10	9	8
TSTCLK1	TSRCLK1	TSTBRG1	BRGDIV	RES	RES	RES	RES
RESET: 0	0	0	0	0	0	0	0

TSTBRG1—Disable BRG1

0 = When the PCMCIA interface is not enabled, the BRG1 clock is driven on the BRG1 pin.
1 = When the PCMCIA interface is not enabled, the BRG1 output is three-stated.

In order to allow the baud rate generator output clocks (which will be driven on TCLK1 and RCLK1 when an internal baud rate generator is used for SCC1) to be disabled, the following bits are provided.

TSRCLK1

0 = RCLK1 is driven on its pin when SCC1 RCLK is the baud rate generator output.
1 = RCLK1 is three-state.

TSTCLK1

0 = TCLK1 is driven on its pin when SCC1 RCLK is the baud rate generator output.
1 = TCLK1 is three-state.

BRGDIV

Enables and disables the divide by two block between the TIN1 pin and the BRG1 prescaler input.

0 = The divide by two block is disabled.
1 = The divide by two block is enabled.

4.3.3 SCC Mode Register (SCM)

Each SCC has a mode register. The functions of bits 5–0 are common to each protocol. The function of the specific mode bits varies according to the protocol selected by the MODE1–MODE0 bits. They are described in the relevant sections for each protocol type. Each SCM is a 16-bit, memory-mapped, read-write register. The SCMs are cleared by reset.

Only the Mode bits have changed functionality. For more information on the other bits, please refer to the *MC68302 Users' Manual*.

15	6	5	4	3	2	1	0
SPECIFIC MODE BITS		DIAG1	DIAG0	ENR	ENT	MODE1	MODE0

DIAG1–DIAG0—Diagnostic Mode

00 = Normal operation ($\overline{\text{CTS}}$, $\overline{\text{CD}}$ lines under automatic control)
01 = Loopback mode
10 = Automatic echo
11 = Software operation (CTS, CD lines under software control)

ENR—Enable Receiver

When ENR is set, the receiver is enabled. When it is cleared, the receiver is disabled, and any data in the receive FIFO is lost. If ENR is cleared during data reception, the receiver aborts the current character. ENR may be set or cleared regardless of whether serial clocks are present. To restart reception, the ENTER HUNT MODE command should be issued before ENR is set again.

ENT—Enable Transmitter

When ENT is set, the transmitter is enabled; when ENT is cleared, the transmitter is disabled. If ENT is cleared, the transmitter will abort any data transmission, clear the transmit data FIFO and shift register, and force the TXD line high (idle). Data already in the transmit shift register will not be transmitted. ENT may be set or cleared regardless of whether serial clocks are present.

MODE1–MODE0—Channel Mode

00 = HDLC
01 = Asynchronous (UART)
10 = **UART 16550 (only on SCC2 when the PCMCIA is enabled)**
11 = BISYNC, Promiscuous Transparent, and **Autobaud**

4.3.4 SCC Data Synchronization Register (DSR)

Each DSR is a 16-bit, memory-mapped, read-write register. DSR specifies the pattern used in the frame synchronization procedure of the SCC in the synchronous protocols. In the UART protocol it is used to configure fractional stop bit transmission. After reset, the DSR defaults to $7E7E (two FLAGs); thus, no additional programming is necessary for the HDLC protocol. For BISYNC the contents of the DSR should be written before the channel is enabled.

15	8	7	0
SYN2		SYN1	

4.3.5 Buffer Descriptors Table

Data associated with each SCC channel is stored in buffers. Each buffer is referenced by a buffer descriptor (BD). BDs are located in each channel's BD table (located in dual-port RAM). There are two such tables for each SCC channel: one is used for data received from the serial line; the other is used to transmit data. The format of the BDs is the same for each SCC mode of operation (HDLC, UART, UART 16550, BISYNC, and transparent) and for both transmit or receive. Only the first field (containing status and control bits) differs for each protocol. The BD format is shown in Figure 4-1.

	15 0
OFFSET + 0	STATUS AND CONTROL
OFFSET + 2	DATA LENGTH
OFFSET + 4	HIGH-ORDER DATA BUFFER POINTER (only lower 8 bits use, upper 8 bits must be 0)
OFFSET + 6	LOW-ORDER DATA BUFFER POINTER

Figure 4-1. SCC Buffer Descriptor Format

NOTE

> Even though the address bus is only 20 bits, the full 32-bit pointer must be Bits 24-32 must be zero, and bits 20-23 are used in the Chip Select address comparison, so they should be programmed to a value which will assert the desired chip select.

4.3.6 SCC Parameter RAM Memory Map

Each SCC maintains a section in the dual-port RAM called the parameter RAM. Each SCC parameter RAM area begins at offset $80 from each SCC base area ($400, $500, or $600) and continues through offset $BF. Part of each SCC parameter RAM (offset $80–$9A), which is identical for each protocol chosen, is shown in Table 4-1. Offsets $9C–$BF comprise the protocol-specific portion of the SCC parameter RAM and are discussed relative to the particular protocol chosen. The SCC parameters have not changed functionality from the MC68302.

Table 4-1. SCC Parameter RAM Memory Map

Address	Name	Width	Description
SCC Base + 80 #	RFCR	Byte	Rx Function Code
SCC Base + 81 #	TFCR	Byte	Tx Function Code
SCC Base + 82 #	MRBLR	Word	Maximum Rx Buffer Length
SCC Base + 84 ##		Word	Rx Internal State
SCC Base + 86 ##		Byte	Reserved
SCC Base + 87 ##	RBD#	Byte	Rx Internal Buffer Number
SCC Base + 88		2 Words	Rx Internal Data Pointer
SCC Base + 8C		Word	Rx Internal Byte Count
SCC Base + 8E		Word	Rx Temp
SCC Base + 90 ##		Word	Tx Internal State
SCC Base + 92 ##		Byte	Reserved
SCC Base + 93 ##	TBD#	Byte	Tx Internal Buffer Number
SCC Base + 94		2 Words	Tx Internal Data Pointer
SCC Base + 98		Word	Tx Internal Byte Count
SCC Base + 9A		Word	Tx Temp
SCC Base + 9C			First Word of Protocol-Specific Area
SCC Base + BF			Last Word of Protocol-Specific Area

# Should be Initialized by the user (M68000 core).
## Modified by the CP following a CP or system reset.

4.3.7 Interrupt Mechanism

The interrupt mechanism for each SCC is the same as the MC68302.

4.3.8 UART Controller

The functionality of the UART controller has not changed. The new Autobaud feature is discussed in 4.3.9 Autobaud Controller (New). For any additional information on parameters, registers, and functionality, please refer to the *MC68302 Users' Manual*.

4.3.8.1 UART MEMORY MAP. When configured to operate in UART mode, the IMP overlays the structure (see Table 4-2) onto the protocol-specific area of that SCC's parameter RAM. Refer to 2.3 System Configuration Registers for the placement of the three SCC parameter RAM areas and to Table 4-1 for the other parameter RAM values.

Table 4-2. UART Specific Parameter RAM

Address	Name	Width	Description
SCC Base + 9C #	MAX_IDL	Word	Maximum IDLE Characters (Receive)
SCC Base + 9E	IDLC	Word	Temporary Receive IDLE Counter
SCC Base + A0 #	BRKCR	Word	Break Count Register (Transmit)
SCC Base + A2 #	PAREC	Word	Receive Parity Error Counter
SCC Base + A4 #	FRMEC	Word	Receive Framing Error Counter
SCC Base + A6 #	NOSEC	Word	Receive Noise Counter
SCC Base + A8 #	BRKEC	Word	Receive Break Condition Counter
SCC Base + AA #	UADDR1	Word	UART ADDRESS Character 1
SCC Base + AC #	UADDR2	Word	UART ADDRESS Character 2
SCC Base + AE	RCCR	Word	Receive Control Character Register
SCC Base + B0 #	CHARACTER1	Word	CONTROL Character 1
SCC Base + B2 #	CHARACTER2	Word	CONTROL Character 2
SCC Base + B4 #	CHARACTER3	Word	CONTROL Character 3
SCC Base + B6 #	CHARACTER4	Word	CONTROL Character 4
SCC Base + B8 #	CHARACTER5	Word	CONTROL Character 5
SCC Base + BA #	CHARACTER6	Word	CONTROL Character 6
SCC Base + BC #	CHARACTER7	Word	CONTROL Character 7
SCC Base + BE #	CHARACTER8	Word	CONTROL Character 8

# Initialized by the user (M68000 core).

4.3.8.2 UART MODE REGISTER . Each SCC mode register is a 16-bit, memory- mapped, read-write register that controls the SCC operation. The read-write UART mode register is cleared by reset.

15	14	13	12	11	10	9	8	7	6	5	0
TPM1	TPM0	RPM	PEN	UM1	UM0	FRZ	CL	RTSM	SL	COMMON SCC MODE BITS	

4.3.8.3 UART RECEIVE BUFFER DESCRIPTOR (RX BD) . The CP reports information about each buffer of received data by its BDs. The Rx BD is shown in Figure 4-2.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET + 0	E	X	W	I	C	A	M	ID	—	—	BR	FR	PR	—	OV	CD
OFFSET + 2	DATA LENGTH															
OFFSET + 4	RX BUFFER POINTER (24-bits used, upper 8 bits must be 0)															
OFFSET +6																

Figure 4-2. UART Receive Buffer Descriptor

4.3.8.4 UART TRANSMIT BUFFER DESCRIPTOR (TX BD) . Data is presented to the CP for transmission on an SCC channel by arranging it in buffers referenced by the channel's Tx BD table. The Tx BD shown in Figure 4-3.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET + 0	R	X	W	I	CR	A	P	—	—	—	—	—	—	—	—	CT
OFFSET + 2	DATA LENGTH															
OFFSET + 4 OFFSET +6	TX BUFFER POINTER (24-bits used, upper 8 bits must be 0)															

Figure 4-3. UART Transmit Buffer Descriptor

4.3.8.5 UART EVENT REGISTER . The SCC event register (SCCE) is called the UART event register when the SCC is operating as a UART.

7	6	5	4	3	2	1	0
CTS	CD	IDL	BRK	CCR	BSY	TX	RX

4.3.8.6 UART MASK REGISTER . The SCC mask register (SCCM) is referred to as the UART mask register when the SCC is operating as a UART. If a bit in the UART mask register is a one, the corresponding interrupt in the event register will be enabled. If the bit is zero, the corresponding interrupt in the event register will be masked. This register is cleared upon reset.

4.3.9 Autobaud Controller (New)

The autobaud function determines the baud rate and format of an asynchronous data stream starting with a known character. This controller may be used to implement the standard AT command set or other characters.

In order to use the autobaud mode, the serial communication controller (SCC) is initially programmed to BISYNC mode. The SCC receiver then synchronizes on the falling edge of the START bit. Once a START bit is detected, each bit received is processed by the autobaud controller. The autobaud controller measures the length of the START bit to determine the receive baud rate and compares the length to values in a user supplied lookup table. After the baud rate is determined, the autobaud controller assembles the character and compares it against two user-defined characters. If a match is detected, the autobaud controller interrupts the host and returns the determined nominal start value from the lookup table. The autobaud controller continues to assemble the characters and interrupt the host until the host stops the reception process. The incoming message should contain a mixture of even and odd characters so that the user has enough information to decide on the proper character format (length and parity). The host then uses the returned nominal start value from the lookup table, modifies the SCC configuration register (SCON) to generate the correct baud rate, and reprograms the SCC to UART mode.

Many rates are supported including: 150, 300, 600, 1200, 2400, 4800, 9600, 14.4K, 19.2K, 38.4K, 57.6K, 64K, 96K, and 115.2K. To estimate the performance of the autobaud mode, the performance table in Appendix A can be used. The maximum full-duplex rate for a

BISYNC channel is one-tenth of the system clock rate. So a 20 MHz IMP can support 115.2K autobaud rate with 2 low-speed channels. The performance can vary depending on system loading, configuration, and echoing mode.

It is important that the highest priority SCC be used for the autobaud function, since it is running at a very high rate.   Any SCC that is guaranteed to be idle during the search operation of the autobaud process will not impact the performance of autobaud in an application. Idle is defined as not having any transmit or receive requests to/from the SCC FIFOs.

4.3.9.1 AUTOBAUD CHANNEL RECEPTION PROCESS. The interface between the autobaud controller and the host processor is implemented with shared data structures in the SCC parameter RAM and in external memory and through the use of a special command to the SCC.

The autobaud controller uses receive buffer descriptor number 7 (Rx BD7) for the autobaud command descriptor. This Rx BD is initialized by the host to contain a pointer to a lookup table residing in the external RAM (contains the maximum and nominal START bit length for each baud rate). The host also prepares two characters against which the autobaud controller will compare the received character (usually these characters are ‘a’ and ‘A’) and the host initializes a pointer to a buffer in external memory where the assembled characters will be stored until the host stops the autobaud process. Finally, the host initializes the SCC data synchronization register (DSR) to $7FFF in order to synchronize on the falling edge of the START bit.

Once the data structures are initialized, the host programs the SCON register to provide a sampling clock that is 16X the maximum supported baud rate. The host then issues the Enter_Baud_Hunt command and enables the SCC in the BISYNC mode.

The autobaud controller reception process begins when the START bit arrives. The autobaud controller then begins to measure the START bit length. With each byte received from the SCC that “belongs” to the START bit, the autobaud controller increments the start length counter and compares it to the current lookup table entry. If the start length counter passed the maximum bit length defined by the current table entry, the autobaud controller switches to the next lookup table entry (the next slower baud rate). This process goes on until the autobaud controller recognizes the end of the START bit. Then, the autobaud controller starts the character assembly process.

The character assembly process uses the nominal bit length, taken from the current lookup table entry, to sample each incoming bit in it’s center. Each bit received is stored to form an 8-bit character. When the assembly process is completed (a STOP bit is received), the character is compared against two user-defined characters.

If the received character does not match any of the two user defined characters, the autobaud controller re-enters the Enter_Baud_Hunt process. The host is not notified until a match is encountered.

If a match is found, the character is written to the received control character register (RCCR) with the corresponding status bit set in Rx BD7. The channel will generate the control character received (CCR) interrupt (bit 3 in the SCCE), if enabled.   If the character matched, but a framing error was detected on the STOP bit, the autobaud controller will also set the framing error status bit in Rx BD7.

The autobaud controller then continues to assemble the incoming characters and to store them in the external data buffer. The host receives a CCR interrupt after each character is received.The host is responsible for determining the end of the incoming message (for example, a carriage return), stopping the autobaud process, and reprogramming the SCC to UART mode. The autobaud controller returns the nominal START bit length value for the detected baud rate from the lookup table and a pointer to the last character received that was written to the external data buffer. The host must be able to handle each character interrupt in order to determine parity and character length (this information may be overwritten when the next character interrupt is presented to the host). The host uses the two received characters to determine 1) whether a properly formed “at” or “AT” was received, and 2) the proper character format (character length, parity).

Once this is decided, three possible actions can result. First, the host may decide that the data received was not a proper “at” or “AT”, and issue the Enter_Baud_Hunt command to cause the autobaud controller to resume the search process. Second, the host may decide the “at” or “AT” is proper and simply continue to receive characters in BISYNC mode. Third, the M68000 core may decide that the “at” or “AT” is proper, but a change in character length or parity is required.

4.3.9.2 AUTOBAUD CHANNEL TRANSMIT PROCESS. The autobaud microcode package supports two methods for transmission. The first method is automatic echo which is supported directly in the SCC hardware, and the second method is a smart echo or software transmit which is supported with an additional clock and software.

Automatic echo is enabled by setting the DIAG bits in the SCC mode register (SCM) to ‘10’ and asserting the $\overline{\text{CD}}$ pin (externally on SCC1 and on SCC2 and SCC3, either externally or by leaving the pin as a general purpose input). The ENT bit of the SCC should remain cleared. The transmitter is not used, so this echoing method does not impact performance.

The smart echo or software transmit requires use of an additional clock and the transmitter, so the overall performance could be affected if other SCCs are running. This method requires an additional clock for sampling the incoming bit stream since the baud rate generator (BRG) must be used to provide the correct frequency for transmission. The user needs to provide the sampling clock that will be used for the autobaud function on the RCLK pin (for example, a 1.8432 MHz clock for 115.2K). The clock that will be used for the SCC transmission can be provided to the BRG from the system clock or on TIN1.The TIN1 and RCLK1 pins can be tied together externally. After the first two characters have been received and character length and parity determined, the host programs the DSR to $FFFF, enables the transmitter (by setting ENT), and programs the transmit character descriptor (overlays CONTROL Character 8). The host is interrupted after each character is transmitted.

For modem applications with the MC68PM302, SCC2 will be used as the DTE interface and autobauding to the DTE baud rate will often be required. If use of the smart echo feature is desired, the receive clock can be provided by the baud rate generator 2 (BRG2) internally by resetting the RCS bit in the SCON2 register to zero. The separate transmit clock can be provided externally to the TCLK2 pin through a hardwire connection to either of the other baud rate generator pin (BRG3). The TCS bit in the SCON2 register should be set to one to enable the external clock source. After autobauding is complete, both the transmit and receive clock sources can be derived internally from BRG2 and the external pin connected to TCLK2 should be three stated to assure that it does not contend with the TCLK2 pin. This can be done by programming the BRG3 pin to a general purpose input by resetting bit 12 in the PACNT register to zero and resetting bit 12 in the PADDR register.

4.3.9.3 AUTOBAUD PARAMETER RAM. When configured to operate in the autobaud mode, the IMP overlays some entries of the UART-specific parameter RAM as illustrated in Table 4-3.

Table 4-3. Autobaud Specific Parameter

Address	Name	Width	Description
SCC Base + 9C *	MAX_IDL	Word	Maximum IDLE Characters
SCC Base + 9E	MAX_BIT	Word	Current Maximum START Bit Length
SCC Base + A0	NOM_START	Word	Current Nom. START Bit (used to determine baud rate)
SCC Base + A2 *	PAREC	Word	Receive Parity Error Counter
SCC Base + A4 *	FRMEC	Word	Receive Framing Error Counter
SCC Base + A6 *	NOSEC	Word	Receive Noise Counter
SCC Base + A8 *	BRKEC	Word	Receive Break Error Counter
SCC Base + AA *	ABCHR1	Word	User Defined Character1
SCC Base + AC *	ABCHR2	Word	User Defined Character2
SCC Base + AE	RCCR	Word	Receive Control Character Register
SCC Base + B0 *	CHARACTER1	Word	CONTROL Character1
SCC Base + B2 *	CHARACTER2	Word	CONTROL Character2
SCC Base + B4 *	CHARACTER3	Word	CONTROL Character3
SCC Base + B6 *	CHARACTER4	Word	CONTROL Character4
SCC Base + B8 *	CHARACTER5	Word	CONTROL Character5
SCC Base + BA *	CHR6/RxPTR	Word	CONTRChar6/MSW of pointer to external Rx Buffer
SCC Base + BC *	CHR7RxPTR	Word	CONTRChar7/LSW of pointer to external Rx Buffer
SCC Base + BE *	CHR8/TxBD	Word	CONTROL Character8/Transmit BD

* These values should be initialized by the user (M68000 core).

Note the new parameters that have been added to the table. They are MAX_BIT, NOM_START, ABCHR1, ABCHR2, RxPTR (2 words), and TxBD. These parameters are of special importance to the autobaud controller. They must be written prior to issuing the Enter_Baud_Hunt command.

When the channel is operating in the autobaud hunt mode, the MAX_BIT parameter is used to hold the current maximum START bit length. The NOM_START location contains the current nominal start from the lookup table. After the autobaud is successful and the first character is matched, the user should use the NOM_START value from the autobaud specific parameter RAM to determine which baud rate from the lookup table was detected. Also the Tx internal data pointer (at offset SCC Base + 94) will point to the last character received into external data buffer.

NOTE

> When the channel is operating in the UART mode, the NOM_START_/BRKCR is used as the break count register and must be initialized before a STOP_TRANSMIT command is issued.

The characters ABCHR1 and ABCHR2 are the autobaud characters that should be searched for by the autobaud controller. Typically these are ‘a’ and ‘A’ (i.e. $0061 and $0041) if using the Hayes command set. These characters must be odd in order for the autobaud controller to correctly determine the length of the START bit. Characters are transmitted and received least significant bit first, so the autobaud controller detects the end of the START bit by the least significant bit of the character being a ‘1’.

The RxPTR is a 2 word location that contains a 32-bit pointer to a buffer in external memory used for assembling the received characters and must be initialized before the Enter_Baud_Hunt command is issued.

NOTE

> Since a length for this external buffer is not given, the user must provide enough space in memory for characters to be assembled and written until the autobaud process is to avoid overwriting other data in memory. This location is not used as the CHARACTER7 value in the control character table until the channel operates in normal UART mode. After reception begins in normal UART mode (i.e. the “a” or “A” is found), this entry is available again as a control character table entry.

The TxBD entry is used as the transmit character descriptor for smart echo or software transmit. This location is not used as the CHARACTER8 value in the control character table until the channel operates in normal UART mode. After reception begins in normal UART mode (i.e. the “a” or “A” is found), this entry is available again as a control character table entry.

4.3.9.4 AUTOBAUD PROGRAMMING MODEL. The following sections describe the details of initializing the autobaud microcode, preparing for the autobaud process, and the memory structures used.

4.3.9.4.1 Preparing for the Autobaud Process. The host begins preparation for the autobaud process with the following steps. Steps 1 and 2 are required if the SCC has been used after reset or after UART mode in order to re-enable the process.

1. Disable the SCC by clearing the ENR and ENT bits. (The host may wish to precede this action with the STOP_TRANSMIT commands to abort transmission in an orderly way).
2. Issue the ENTER_HUNT_MODE command to the SCC (This ensures that an open buffer descriptor is closed).
3. Set up all the autobaud parameters in the autobaud specific parameter RAM shown in

Table 4-3, the autobaud command descriptor shown in Table 4-4, and the lookup table shown in Table 4-4. Of these three areas, the autobaud controller only modifies the autobaud specific parameter RAM and the first word of the autobaud command descriptor during its operation.

4. Write the SCON to configure the SCC to use the baud rate generator clock of 16x the maximum supported baud rate. A typical value is $4000 assuming a 1.8432 MHz clock rate on TIN1 and a maximum baud rate of 115.2K, but this can change depending on the maximum baud rate and the EXTAL frequency.
5. Write the DSR of the SCC with the value $7FFF in order to detect the START bit.
6. The host initiates the autobaud search process by issuing the Enter_Baud_Hunt command
7. Write the SCM of the SCC with $1133 to configure it for BISYNC mode, with the REVD and RBCS bits set, software operation mode, and the transmitter disabled. After a few characters have been received, the transmitter can be enabled, and the software echo function may be performed after issuing the RESTART TRANSMIT command.

In general, the autobaud controller uses the same data structure as that of the UART controller. The first character (if matched) is stored in the receiver control character register and the external data buffer, and the status of that character is reported in the autobaud command descriptor. After the first character, each incoming character is then stored in the buffer pointed to by RxPTR, and the status is updated in the autobaud command autobaud descriptor. The Tx internal data pointer (at offset SCC Base + 94) is updated to point to the last character stored in the external data buffer.

4.3.9.4.2 Enter_Baud_Hunt Command. This command instructs the autobaud controller to begin searching for the baud rate of a user predefined character. Prior to issuing the command the M68000 prepares the autobaud command descriptor to contain the lookup table size and pointer.

The Enter_Baud_Hunt uses the GCI command with opcode = 10, and the channel number set for the corresponding SCC. For example, with SCC1, the value written to the command register would be $61.

4.3.9.4.3 Autobaud Command Descriptor. The autobaud controller uses the receive buffer descriptor number 7 (Rx BD7) as an autobaud command descriptor. The autobaud command descriptor is used by the M68000 core to transfer command parameters to the autobaud controller, and by the autobaud controller to report information concerning the received character.

The structure of the autobaud command descriptor for the autobaud process is shown in Table 4-4. The first word of the descriptor or the status word is updated after every character is received.

Offset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0						FE	M2	M1					EOT		OV	CD
2	LOOKUP TABLE SIZE															
4	FUNCTION CODE															
8	LOOKUP TABLE POINTER															

Table 4-4. Autobaud Command Descriptor

FE – Framing Error (Bit 10)

If this bit is set, a character with a framing error was received. A framing error is detected by the autobaud controller when no STOP bit is detected in the received data. FE will be set for a 9-bit character (8 bits + parity) if the parity bit is ‘0’.

NOTE

The user must clear this bit when it is set.

M2 – Match Character2 (Bit 9)

When this bit is set, the character received matched the User Defined Character 2. The received character is written into the receive control character register (RCCR).

M1 – Match Character1 (Bit 8)

When this bit is set, the character received matched the User Defined Character 1. The received character is written into the receive control character register (RCCR).

EOT – End Of Table (bit 3)

When this bit is set, the autobaud controller measured start length exceeded the maximum start length of the last entry in the lookup table (lowest baud rate).

NOTE

The user must clear this bit when it is set.

OV – Overrun (bit 1)

If this bit is set, a receiver overrun occurred during autobaud reception.

NOTE

The user must clear this bit when it is set.

CD – Carrier Detect Lost (bit 0)

If this bit is set, the carrier detect signal was negated during autobaud reception.

NOTE

The user must clear this bit when it is set.

Lookup Table Size - Lookup table size is the number of baud rate entries in the external lookup table.

Lookup Table Pointer - The lookup table pointer is the address in the external RAM where the lookup table begins.

NOTE

The lookup table cannot cross a 64k memory block boundary.

4.3.9.4.4 Autobaud Lookup Table. The autobaud controller uses an external lookup table to determine the baud rate while in the process of receiving a character. The lookup table contains two entries for each supported baud rate. The first entry is the maximum start length for the particular baud rate, and the second entry is the nominal length for a 1/2 START bit.

To determine the two values for each table entry, first calculate the autobaud sampling rate (EQ 2). To do this EQ 1 must be used until EQ 2 is satisfied. The sampling rate is the lowest speed baud rate that can be generated by the SCC baud rate generator that is over a threshold defined in EQ 2.

$$\text{BRG Clk Rate} = \text{System Clock or TIN1} / ((\text{Clock Divider bits in SCON}) + 1) \qquad \text{(EQ 1)}$$

assuming that the DIV bit in SCON is set to 0, (otherwise an additional “divide-by-4” must be included).

$$\text{Sampling Rate} = \text{BRG Clk Rate, where BRG Clk rate} >= (\text{Max Desired UART Baud Rate}) \text{ x } 16 \qquad \text{(EQ 2)}$$

For instance, if a 115.2K baud rate is desired, with a 16.67 MHz system clock, the minimum sampling rate possible is 1.843 MHz = 115.2K x 16. This exact frequency can be input to RCLK1 or TIN1 as the sample clock. If the system clock is to be used, a 16.67 MHz system clock cannot produce an exact baud rate clock of 1.843 MHz. The lowest one that can be used is Baud Rate = 16.67 MHz / (7+1) = 2.083 MHz. Thus, 2.083 MHz is the sampling rate, and the SCON should be set to $000E to produce this.

Once the sampling rate is known, the other two equations follow easily. The maximum START bit length is calculated by the following equation:

$$\text{Maximum start length} = (\text{Sampling Rate/Recognized baud rate}) \text{ x } 1.05 \qquad \text{(EQ 3)}$$

Thus, for the first entry in the table, the maximum start length is 1.8432 Mhz/115200 x 1.05 = 17 for an external sample clock. The value 1.05 is a suggested margin that allows characters 5% larger than the nominal character rate to be accepted. In effect, the margin determines the “split point” between what is considered to be a 56.7K character rate and what is a 38.4K character rate. The margin should not normally be less than 1.03 due to clocking differences between UARTs.

The nominal START bit length is calculated by:

$$\text{Nominal start length} = (\text{Sampling Rate/Recognized baud rate}) / 2 \qquad \text{(EQ 4)}$$

For the 115.2K example in the first table entry, this would be 1.8432 MHz/115.2K/2 = 8.

The structure of the lookup table is shown in Table 4-5. The table starts with the maximum UART baud rate supported and ends with the minimum UART baud rate supported.

Table 4-5. Autobaud Lookup Table Format

OFFSET from Lookup Table Pointer	DESCRIPTION
0	Maximum Start Length
2	Nominal Start Length
4	Maximum Start Length
6	Nominal Start Length
•	Maximum Start Length
•	Nominal Start Length
(Lookup Table Size - 1) * 4	Maximum Start Length
[(Lookup Table Size - 1) * 4] + 2	Nominal Start Length

NOTE

If less margin is used in the calculation of the maximum start length above, it is possible to distinguish between close UART rates such as 64K and 57.6K. However variations in RS232 drivers of up to 4%, plus nominal clocking rate variations of 3%, plus the fact that the sampling rate may not perfectly divide into the desired UART rate, can make this distinction difficult to achieve in some scenarios.

4.3.9.5 LOOKUP TABLE EXAMPLE.

Table 4-6 is an example autobaud lookup table. The maximum start and nominal start values are derived assuming a 1.8432 MHz sampling clock on TIN1 or RCLK and a shift factor of 5%.

Table 4-6. Lookup Table Example

Desired Baud Rate	Maximum Start	Nominal Start
115200	19	8
57600	36	16
38400	54	24
28800	6a	32
19200	10a	48
14400	13a	64
12000	166	77
9600	208	96
7200	270	128
4800	410	192
2400	810	384
1200	1630	768
600	3240	1536
300	6470	3072
110	17600	8378

4.3.9.6 DETERMINING CHARACTER LENGTH AND PARITY. Table 4-7 shows the different possible character lengths and parity that will be discussed. The following paragraphs will discuss for each case how to determine the parity.

Table 4-7. Character Lengths and Parity Cases

Case #	Character Length	Parity	Notes
1	7-bit	No parity, 1 STOP bit	Not Supported

2	7-bit	Even parity
		Odd parity
		Parity=1
		Parity=0
			Parity is indicated by the most significant bit of the byte

3	8-bit	No parity	Same as 7-bit, parity=0
4	8-bit	Even parity Odd parity Parity=0	Parity is indicated by which characters generate a FE interrupt
5	8-bit	Parity=1	Not Supported

- Case 1– This case cannot be supported because the autobaud can not separate the first character from the second character.
- Case 2– As each character is assembled, it is stored into a complete byte. Assuming that the characters are ASCII characters with 7-bit codes, the 8th bit of the byte will contain the parity bit. If the parity is either even or odd, then after receiving an odd character and an even character, the 8th bit should be different for the odd and even characters. The parity can be determined by the setting of the parity bit for one of the two characters. If the 8th bit is always a 1, this is the same as a 7-bit character, no parity and at least 2 STOP bits or a 7-bit character with force 1 parity. If the 8th bit is always a zero, then either the character is a 7-bit character with force 0 parity, or the character is a 8-bit character with no parity.
- Case 3– This case is the same as 7-bit character with force 0 parity. The 8th bit of the byte will always be zero.
- Case 4–This case assumes a 8-bit character with the 8th bit of the character equal to a 0 (ASCII character codes define the 8th bit as zero). If the parity is either even or odd, then after receiving an odd and an even character, a framing error (FE) interrupt should have been generated for one of them (the interrupt is generated when the parity bit is zero). The user can determine the parity by which character generated a FE interrupt (if the odd character did, then the parity is odd). If a framing error occurs on every character, then the character is 8-bits with force 0 parity. If no framing error occurs, than this is the same as Case 5.
- Case 5– This case is not supported, because it can not be differentiated from 7-bit force 0 parity and 8-bit no parity. If the 9th bit is a 1, then it will be interpreted as a STOP bit.

4.3.9.7 AUTOBAUD RECEPTION ERROR HANDLING PROCEDURE. The autobaud controller reports reception error conditions using the autobaud command descriptor. Three types of errors are supported:

- Carrier Detect Lost during reception

When this error occurs and the channel is not programmed to control this line with software, the channel terminates reception, sets the carrier detect lost (CD) bit in the command descriptor, and generates the CCR interrupt, if enabled. CCR is bit 3 of the SCCE register.

- Overrun Error

When this error occurs, the channel terminates reception, sets the overrun (OV) bit in the command descriptor, and generates the CCR interrupt, if enabled.

- End Of Table Error

When this error occurs, the channel terminates reception, sets the end of table (EOT) bit in the command descriptor, and generates the CCR interrupt, if enabled.

Any of these errors will cause the channel to abort reception. In order to resume autobaud operation after an error condition, the M68000 should clear the status bits and issue the Enter_Baud_Hunt command again.

4.3.9.8 AUTOBAUD TRANSMISSION. The autobaud package supports two methods for echoing characters or transmitting characters. The two methods are automatic echo and smart echo.

4.3.9.8.1 Automatic Echo. This method uses the SCC hardware to automatically echo the characters back on the TxD pin. The automatic echo is enabled by setting the DIAG bits in the SCM to ‘10’. The transmitter should not be enabled. The hardware echo is done automatically. The $\overline{\text{CD}}$ pin needs to be asserted in order for the characters to be transmitted back. On SCC1, the external $\overline{\text{CD}}$ pin must be tied low. On SCC2 and SCC3, either the external $\overline{\text{CD}}$ pin must be tied low or the $\overline{\text{CD}}$ pins should be left configured as general purpose input pins (the $\overline{\text{CD}}$ signal to the SCC is then connected to ground internally).

Using the automatic echo, the receiver still autobauds correctly and performance is not affected. The SCC echoes the received data with a few nanoseconds delay.

4.3.9.8.2 Smart Echo. This method requires addition hardware and software to implement. The user must provide two clock sources. One clock source is the sample clock which is input on RCLK and cannot be divided down. The BRG is used to divide the second clock down to provide the clock used for transmit. The second clock can be either the system clock or a clock connected to TIN1. The TIN1 and RCLK pins can be connected to each other externally.

After the first character is received, the user must take the following steps:

1. Determine the baud rate from the returned NOM_START value and program SCON to (input frequency/baud rate)-1, where the input frequency is either the system clock or the clock on TIN1.
2. Program the DSR to $FFFF. The DSR will need to be programmed back to $7FFF before the Enter_Baud_Hunt command is issued again.
3. Set the ENT bit in the mode register.

4. Program the transmit character BD as show in Table 4-8.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R		CL		PE	PM			CHAR							

Table 4-8. Transmit Character BD

R (Ready Bit)

0 = Character is not ready
1 = Character is ready to transmit

CL (Character Len)

0 = 7 bits + parity or 8 bits with no parity
1 = 8 bits + parity

PE (Parity Enable)

0 = No parity
1 = Parity

PM (Parity Mode)

0 = Even parity
1 = Odd parity

The autobaud controller issues a Tx interrupt after each character is transmitted.

4.3.9.9 REPROGRAMMING TO UART MODE OR ANOTHER PROTOCOL. The following steps should be followed in order to switch the SCC from autobaud to UART mode or to another protocol.

- Disable the SCC by clearing ENR and ENT.
- Issue the Enter_Hunt_Mode command.
- Initialize the SCC parameter RAM (specifically, the Rx and Tx internal states and the words containing the Rx and Tx BD#s) to the state immediately after reset and initialize the protocol specific parameter area for the new protocol.
- Re-enable the SCC with the new mode.

4.3.10 HDLC Controller

The functionality of the HDLC controller has not changed. For any additional information on parameters, registers, and functionality, please refer to the *MC68302 Users' Manual*.

4.3.10.1 HDLC MEMORY MAP. When configured to operate in HDLC mode, the IMP overlays the structure shown in Table 4-9 onto the protocol-specific area of that SCC parameter RAM. Refer to System Configuration Registers on page 4 for the placement of the three SCC parameter RAM areas and to Table 4-1 for the other parameter RAM values.

Table 4-9. HDLC-Specific Parameter RAM

Address	Name	Width	Description
SCC Base + 9C	RCRC_L	Word	Temp Receive CRC Low
SCC Base + 9E	RCRC_H	Word	Temp Receive CRC High
SCC Base + A0 #	C_MASK_L	Word	Constant ($F0B8 16-Bit CRC, $DEBB 32-Bit CRC)
SCC Base + A2 #	C_MASK_H	Word	Constant ($XXXX 16-Bit CRC, $20E3 32-Bit CRC)
SCC Base + A4	TCRC_L	Word	Temp Transmit CRC Low
SCC Base + A6	TCRC_H	Word	Temp Transmit CRC High
SCC Base + A8 #	DISFC	Word	Discard Frame Counter
SCC Base + AA #	CRCEC	Word	CRC Error Counter
SCC Base + AC #	ABTSC	Word	Abort Sequence Counter
SCC Base + AE #	NMARC	Word	Nonmatching Address Received Counter
SCC Base + B0 #	RETRC	Word	Frame Retransmission Counter
SCC Base + B2 #	MFLR	Word	Max Frame Length Register
SCC Base + B4	MAX_cnt	Word	Max_Length Counter
SCC Base + B6 #	HMASK	Word	User-Defined Frame Address Mask
SCC Base + B8 #	HADDR1	Word	User-Defined Frame Address
SCC Base + BA #	HADDR2	Word	User-Defined Frame Address
SCC Base + BC #	HADDR3	Word	User-Defined Frame Address
SCC Base + BE #	HADDR4	Word	User-Defined Frame Address

# Should be initialized by the user (M68000 core).

4.3.10.2 HDLC MODE REGISTER . Each SCC mode register is a 16-bit, memory-mapped, read-write register that controls the SCC operation. The term HDLC mode register refers to the protocol-specific bits (15–6) of the SCC mode register when that SCC is configured for HDLC. The read-write HDLC mode register is cleared by reset.

15	14	13	12	11	10	9	8	7	6	5	0
NOF3	NOF2	NOF1	NOF0	C32	FSE	—	RTE	FLG	ENC	COMMON SCC MODE BITS	

4.3.10.3 HDLC RECEIVE BUFFER DESCRIPTOR (RX BD). The HDLC controller uses the Rx BD to report information about the received data for each buffer. The Rx BD is shown in Figure 4-4.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET + 0	E	X	W	I	L	F	—	—	—	—	LG	NO	AB	CR	OV	CD
OFFSET + 2	DATA LENGTH															
OFFSET + 4 OFFSET +6	RXBUFFER POINTER (24-bits used, upper 8 bits must be 0)															

Figure 4-4. HDLC Receive Buffer Descriptor

4.3.10.4 HDLC TRANSMIT BUFFER DESCRIPTOR (TX BD) . Data is presented to the HDLC controller for transmission on an SCC channel by arranging it in buffers referenced by the channel's Tx BD table. The HDLC controller confirms transmission (or indicates error conditions) using the BDs to inform the M68000 core that the buffers have been serviced. The Tx BD is shown in Figure 4-5.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET + 0	R	X	W	I	L	TC	—	—	—	—	—	—	—	—	UN	CT
OFFSET + 2	DATA LENGTH															
OFFSET + 4	TXBUFFER POINTER (24-bits used, upper 8 bits must be 0)															
OFFSET + 6																

Figure 4-5. HDLC Transmit Buffer Descriptor

4.3.10.5 HDLC EVENT REGISTER . The SCC event register (SCCE) is called the HDLC event register when the SCC is operating as an HDLC controller. It is an 8-bit register used to report events recognized by the HDLC channel and to generate interrupts. Upon recognition of an event, the HDLC controller sets its corresponding bit in the HDLC event register. Interrupts generated by this register may be masked in the HDLC mask register.This register is cleared at reset.

7	6	5	4	3	2	1	0
CTS	CD	IDL	TXE	RXF	BSY	TXB	RXB

4.3.10.6 HDLC MASK REGISTER. The SCC mask register (SCCM) is referred to as the HDLC mask register when the SCC is operating as an HDLC controller. It is an 8-bit read-write register that has the same bit formats as the HDLC event register. If a bit in the HDLC mask register is a one, the corresponding interrupt in the event register will be enabled. If the bit is zero, the corresponding interrupt in the event register will be masked. This register is cleared upon reset.

4.3.11 BISYNC Controller

The functionality of the BISYNC controller has not changed. For any additional information on parameters, registers, and functionality, please refer to the *MC68302 Users' Manual*.

4.3.11.1 BISYNC MEMORY MAP . When configured to operate in BISYNC mode, the IMP overlays the structure listed in Table 4-10 onto the protocol-specific area of that SCC parameter RAM. Refer to System Configuration Registers on page 4 for the placement of the three SCC parameter RAM areas and Table 4-1 for the other parameter RAM values.

Table 4-10. BISYNC Specific Parameter RAM

Address	Name	Width	Description
SCC Base + 9C	RCRC	Word	Temp Receive CRC
SCC Base + 9E	CRCC	Word	CRC Constant
SCC Base + A0 #	PRCRC	Word	Preset Receiver CRC 16/LRC
SCC Base + A2	TCRC	Word	Temp Transmit CRC
SCC Base + A4 #	PTCRC	Word	Preset Transmitter CRC 16/LRC
SCC Base + A6	RES	Word	Reserved
SCC Base + A8	RES	Word	Reserved
SCC Base + AA #	PAREC	Word	Receive Parity Error Counter
SCC Base + AC #	BSYNC	Word	BISYNC SYNC Character
SCC Base + AE #	BDLE	Word	BISYNC DLE Character
SCC Base + B0 #	CHARACTER1	Word	CONTROL Character 1
SCC Base + B2 #	CHARACTER2	Word	CONTROL Character 2
SCC Base + B4 #	CHARACTER3	Word	CONTROL Character 3
SCC Base + B6 #	CHARACTER4	Word	CONTROL Character 4
SCC Base + B8 #	CHARACTER5	Word	CONTROL Character 5
SCC Base + BA #	CHARACTER6	Word	CONTROL Character 6
SCC Base + BC #	CHARACTER7	Word	CONTROL Character 7
SCC Base + BE #	CHARACTER8	Word	CONTROL Character 8

# Should be initialized by the user (M68000 core).

4.3.11.2 BISYNC MODE REGISTER. Each SCC mode register is a 16-bit, memory-mapped, read-write register that controls the SCC operation. The term BISYNC mode register refers to the protocol-specific bits (15–6) of the SCC mode register when that SCC is configured for BISYNC. The read-write BISYNC mode register is cleared by reset.

15	14	13	12	11	10	9	8	7	6	5 0
PM	EXSYN	NTSYN	REVD	BCS	—	RTR	RBCS	SYNF	ENC	COMMON SCC MODE BITS

4.3.11.3 BISYNC RECEIVE BUFFER DESCRIPTOR (RX BD). The CP reports information about the received data for each buffer using BD. The Rx BD is shown in Figure 4-6.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET + 0	E	X	W	I	C	B	—	—	—	—	—	DL	PR	CR	OV	CD
OFFSET +2	DATA LENG															
OFFSET +4	RXBUFFER POINTER (24-bits used, upper 8 bits must be 0)															
OFFSET + 6																

Figure 4-6. BISYNC Receive Buffer Descriptor

4.3.11.4 BISYNC TRANSMIT BUFFER DESCRIPTOR (TX BD). Data is presented to the CP for transmission on an SCC channel by arranging it in buffers referenced by the channel's Tx BD table. The Tx BD is shown in Figure 4-7.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET + 0	R	X	W	I	L	TB	B	BR	TD	TR	—	—	—	—	UN	CT
OFFSET + 2	DATA LENGTH															
OFFSET+4	TXBUFFER POINTER (24-bits used, upper 8 bits must be 0)															
OFFSET +6																

Figure 4-7. BISYNC Transmit Buffer Descriptor

4.3.11.5 BISYNC EVENT REGISTER. The SCC event register (SCCE) is referred to as the BISYNC event register when the SCC is programmed as a BISYNC controller. It is an 8-bit register used to report events recognized by the BISYNC channel and to generate interrupts. On recognition of an event, the BISYNC controller sets the corresponding bit in the BISYNC event register. Interrupts generated by this register may be masked in the BISYNC mask register. This register is cleared at reset.

7	6	5	4	3	2	1	0
CTS	CD	—	TXE	RCH	BSY	TX	RX

4.3.11.6 BISYNC MASK REGISTER. The SCC mask register (SCCM) is referred to as the BISYNC mask register when the SCC is operating as a BISYNC controller. It is an 8-bit read-write register that has the same bit format as the BISYNC event register. If a bit in the BISYNC mask register is a one, the corresponding interrupt in the event register will be enabled. If the bit is zero, the corresponding interrupt in the event register will be masked. This register is cleared upon reset.

4.3.12 Transparent Controller

The functionality of the transparent controller has not changed. For any additional information on parameters, registers, and functionality, please refer to the *MC68302 Users' Manual*.

4.3.12.1 TRANSPARENT MEMORY MAP. When configured to operate in transparent mode, the IMP overlays the structure illustrated in Table 4-11 onto the protocol specific area of that SCC parameter RAM. Refer to System Configuration Registers on page 4 for the placement of the three SCC parameter RAM areas and Table 4-1 for the other parameter RAM values.

Table 4-11. Transparent-Specific Parameter RAM

Address	Name	Width	Description
SCC BASE + 9C	RES	WORD	Reserved
SCC BASE + 9E	RES	WORD	Reserved
SCC BASE + A0	RES	WORD	Reserved
SCC BASE + A2	RES	WORD	Reserved
SCC BASE + A4	RES	WORD	Reserved
SCC BASE + A6	RES	WORD	Reserved
SCC BASE + A8	RES	WORD	Reserved
SCC BASE + AA	RES	WORD	Reserved
SCC BASE + AC	RES	WORD	Reserved
SCC BASE + AE	RES	WORD	Reserved
SCC BASE + B0	RES	WORD	Reserved
SCC BASE + B2	RES	WORD	Reserved
SCC BASE + B4	RES	WORD	Reserved
SCC BASE + B6	RES	WORD	Reserved
SCC BASE + B8	RES	WORD	Reserved
SCC BASE + BA	RES	WORD	Reserved
SCC BASE + BC	RES	WORD	Reserved
SCC BASE + BE	RES	WORD	Reserved

4.3.12.2 TRANSPARENT MODE REGISTER. Each SCC mode register is a 16-bit, memory-mapped, read-write register that controls the SCC operation. The term transparent mode register refers to the protocol-specific bits (15–6) of the SCC mode register when that SCC is configured for transparent mode. The transparent mode register is cleared by reset. All undefined bits should be written with zero.

15	14	13	12	11	10	9	8	7	6	5 0
—	EXSYN	NTSYN	REVD	—	—	—	—	—	—	COMMON SCC MODE BITS

4.3.12.3 TRANSPARENT RECEIVE BUFFER DESCRIPTOR (RXBD). The CP reports information about the received data for each buffer using BD. The RxBD is shown in Figure 4-8.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET + 0	E	X	W	I	—	—	—	—	—	—	—	—	—	—	OV	CD
OFFSET + 2	DATA LENGTH															
OFFSET + 4	RXBUFFER POINTER (24-bits used, upper 8 bits must be 0)															
OFFSET +6																

Figure 4-8. Transparent Receive Buffer Descriptor

4.3.12.4 TRANSPARENT TRANSMIT BUFFER DESCRIPTOR (TX BD). Data is presented to the CP for transmission on an SCC channel by arranging it in buffers referenced by the channel's Tx BD table. The Tx BD is shown in Figure 4-9.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET + 0	R	X	W	I	L	—	—	—	—	—	—	—	—	—	UN	CT
OFFSET + 2	DATA LENGTH															
OFFSET + 4	TXBUFFER POINTER (24-bits used, upper 8 bits must be 0)															
OFFSET +6																

Figure 4-9. Transparent Transmit Buffer Descriptor

4.3.12.5 TRANSPARENT EVENT REGISTER . The SCC event register (SCCE) is referred to as the transparent event register when the SCC is programmed as a transparent controller. It is an 8-bit register used to report events recognized by the transparent channel and to generate interrupts. On recognition of an event, the transparent controller sets the corresponding bit in the transparent event register. Interrupts generated by this register may be masked in the transparent mask register. This register is cleared at reset.

7	6	5	4	3	2	1	0
CTS	CD	—	TXE	RCH	BSY	TX	RX

4.3.12.6 TRANSPARENT MASK REGISTER. The SCC mask register (SCCM) is referred to as the transparent mask register when the SCC is operating as a transparent controller. It is an 8-bit read-write register that has the same bit format as the transparent event register. If a bit in the transparent mask register is a one, the corresponding interrupt in the event register will be enabled. If the bit is zero, the corresponding interrupt in the event register will be masked. This register is cleared at reset.

4.4 16550 EMULATION CONTROLLER (NEW FEATURE)

4.4.1 16550 Emulation Controller Features

- Contains PC port side that implements the full register set, external pins, external pin timing, and interrupt structure of the 16550.
- The 16550 UART registers are accessible from the PCMCIA port in any page of I/O space.
- The emulation controller can optionally scale the transfer rates to and from the PC to match the speeds of a real 16550.
- The interface to the 68000 core maintains the standard 68302 programming model of a normal SCC in UART mode.
- Data is transferred between the 68000 bus and the 16550 emulation controller using SDMA transfers under RISC control.
- Data transfers are never serialized in the emulation controller.

4.4.2 16550 Emulation Controller Overview

The 16550 emulation controller has the complete register set of the 16550 with the same FIFO and interrupt structure. The data transfer between the PC and the 68000 core consists of parallel transfers of data. There is no parallel to serial (shift-register) transformation of the data.

The 16550 emulation controller transfers modem status written by the PC via hardware registers visible to the 68000 core. The 68000 core transfers data and modem status bits to the 16550 emulation controller though register visible to both the 68000 core and the PC. Data transfers to and from the 16550 FIFO or holding registers are routed to IMP memory using the RISC controller and the SDMA channels. Data transfers to and from the PC take place through the PCMCIA interface. The 16550 emulation controller uses SCC2 parameter RAM space and its baud rate generator (BRG) when enable and therefore the SCC2 serial channel cannot be used simultaneously with the 16550 emulation. The 16550 controller uses the same data structures as a typical IMP SCC in UART mode and supports multi-buffer operation.

Two 4-bit counters and a BRG control the data transfer rate of the 16550 emulation controller, the FIFO emulation time-out, and the 302 idle emulation.

4.4.2.1 16550 EMULATION CONTROLLER FIFOS OVERVIEW. The 16550 controller has separate transmit and receive FIFO's for implementing 16550 mode which is enabled by setting the FIFO enable bit in the FIFO control register. Figure 4-11 shows the receiver FIFO which transfers data from the CP 16550 transmit buffer descriptors via the communication processor's peripheral bus to the PC via the PCMCIA bus. The FIFO is 16 bytes deep and includes three error bits that are attached to each byte and are decoded in the line status register as each byte reaches the top of the FIFO. The data is never serialized in the process. The baud rate clock is provided by BRG2 and can be divided down further by a 4 bit counter. The 4 bit counter is useful to slow the baud rate clock down to a rate that emulates the serial transfer rates of a true 16550 while allowing the baud rate generator itself to be programmed with traditional values used for serial UART rates. This will keep the receive interrupt rate to the PC at the level it is used to handling and prevent the PC from being inundated with bursts of fast interrupts.

4.4.3 PC Accesses

The PC accesses the 16550 emulation registers through the PCMCIA interface using I/O space accesses. These read and write cycles are synchronized with the IMP clock. The IMP clock rate affects the synchronization time and the cycle length. When using low rate clocks, $\overline{\text{WAIT}}$ should be asserted on the PCMCIA cycle to delay the completion of the cycle. This can be done by making sure that the NWAIT bit in the 16550 emulation mode register is reset.

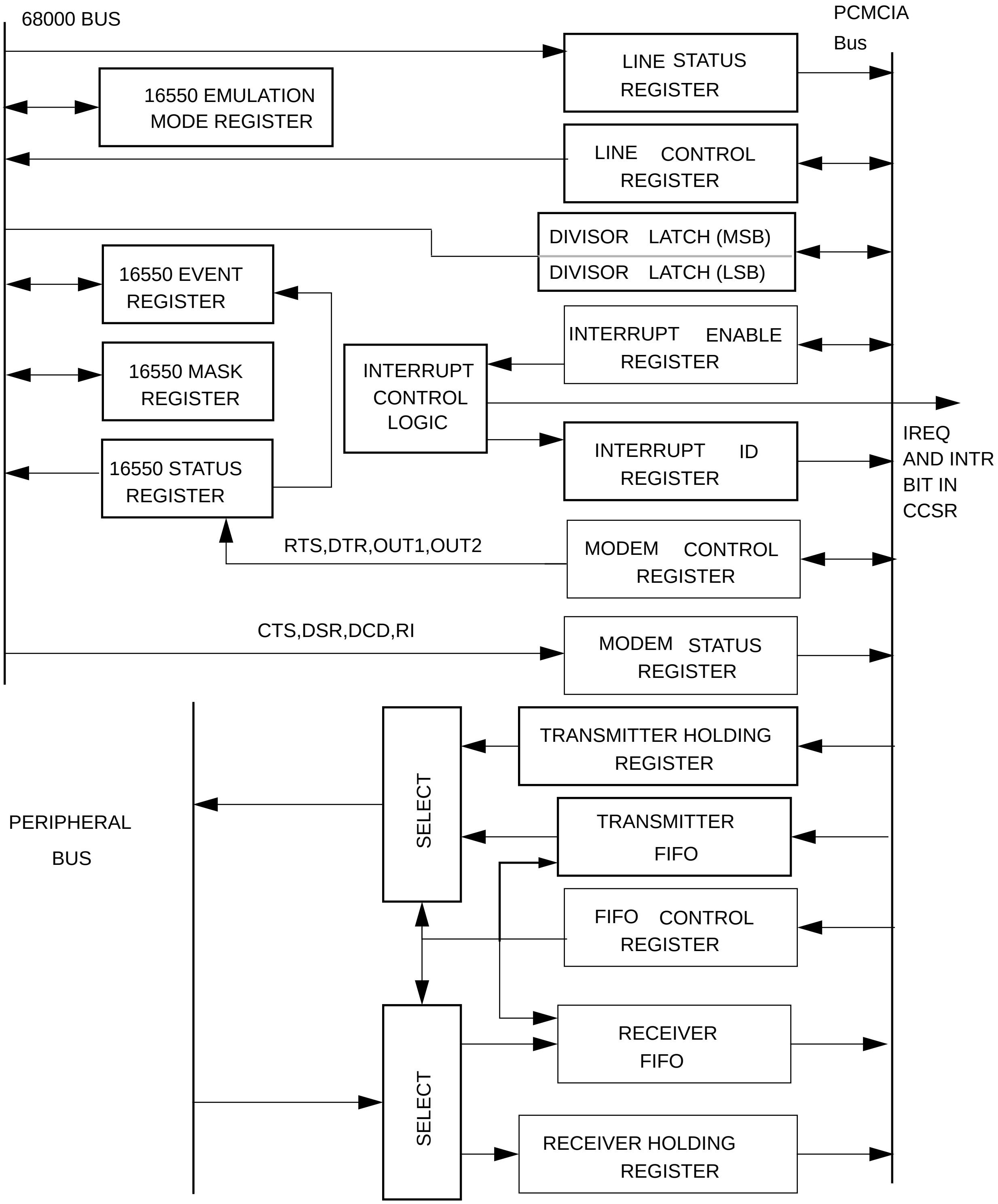


Figure 4-10. 16550 Emulation Controller Block Diagram

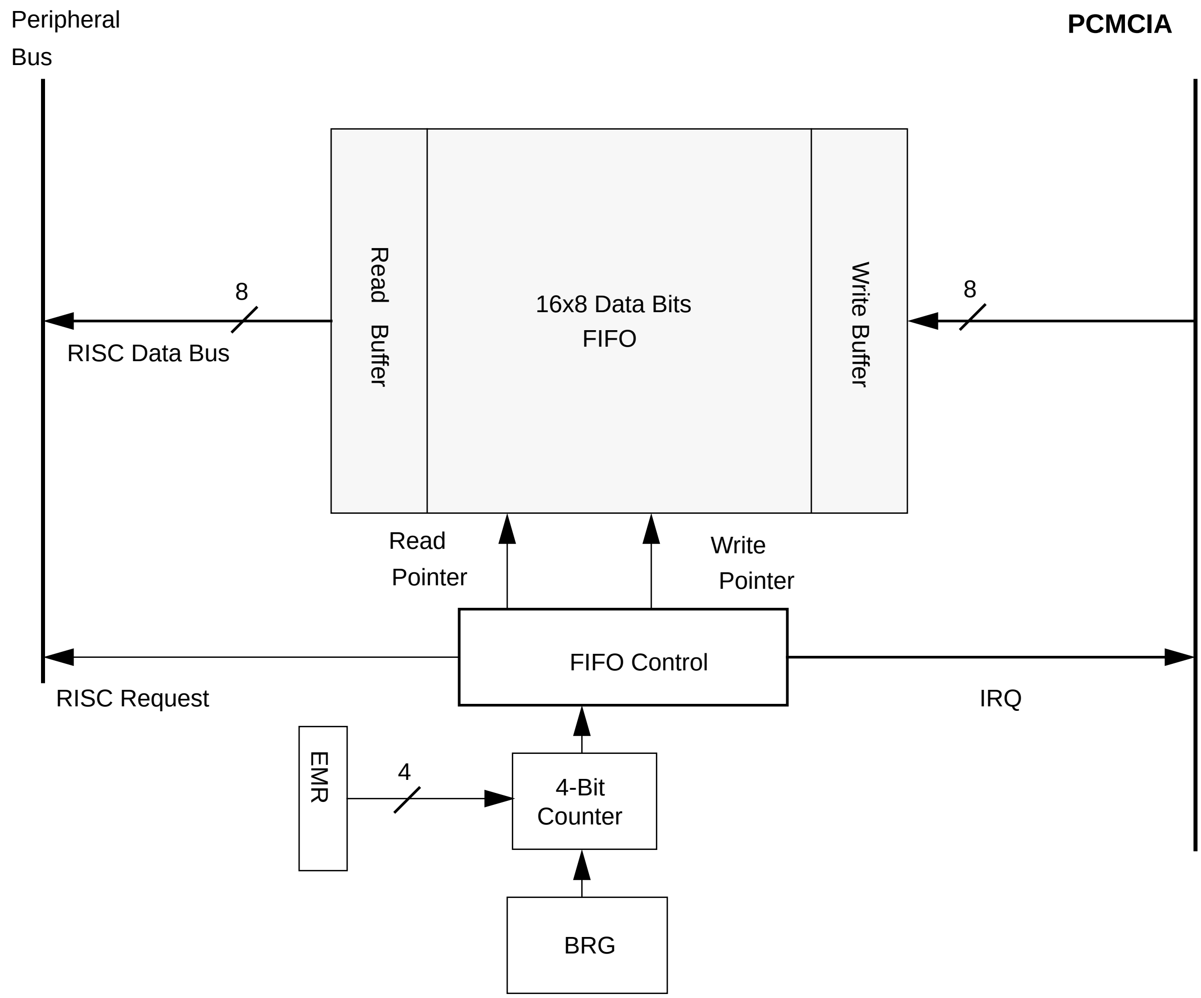


Figure 4-11. 16550 Emulation Controller Transmitter FIFO

4.4.4 PC Programmer Model

The PC programmer model is identical to the standard 16550. All software packages that support the 16650 can be executed without any modification.

Table 4-12. 16550 Emulation Registers Addresses

REG	CE1	IORD	IOWR	DLAB	A25	A2	A1	A0	Register
L	L	L	H	0	0	0	0	0	Receiver buffer (read)
L	L	H	L	0	0	0	0	0	Transmitter holding register (write)
L	L	L	H	0	0	0	0	1	Interrupt enable (read)
L	L	H	L	0	0	0	0	1	Interrupt enable (write)
L	L	L	H	x	0	0	1	0	Interrupt identification (read)
L	L	H	L	x	0	0	1	0	FIFO control (write)
L	L	L	H	x	0	0	1	1	Line control (read)
L	L	H	L	x	0	0	1	1	Line control (write)
L	L	L	H	x	0	1	0	0	MODEM control (read)
L	L	H	L	x	0	1	0	0	MODEM control (write)
L	L	L	H	x	0	1	0	1	Line status (read)
L	L	L	H	x	0	1	1	0	MODEM status (read)
L	L	L	H	x	0	1	1	1	Scratch (read)
L	L	H	L	x	0	1	1	1	Scratch (write)
L	L	L	H	1	0	0	0	0	Divisor latch (LS) (read)
L	L	H	L	1	0	0	0	0	Divisor latch (LS) (write)
L	L	L	H	1	0	0	0	1	Divisor latch (MS) (read)
L	L	H	L	1	0	0	0	1	Divisor latch (MS) (write)

4.4.4.1 16550 EMULATION REGISTERS DESCRIPTION.

4.4.4.1.1 Line Control Register (LCR). This register is used to specify the format of the asynchronous serial data. The 68000 core receives an interrupt when the PC writes to this register (See 16550 Event Register on page 45 for more details). The 68000 may read the register bits and emulate the UART functions.

The PC can write and read the contents of this register. The 68000 can only read this register.

LCR Base + $8F2

7	6	5	4	3	2	1	0
DLAB	SET BREAK	STICK PARITY	EVEN PARITY	PARITY ENABLE	STOP BIT LENGTH	CHARACTER LENGTH	
RESET 0	O	0	0	0	0	0	0

Read/Write—PC
Read-Only—68000

Bits 0-5

These bits do not affect the 16550 emulation controller. The 68000 core may read them and use them in the emulation process. The 16550 emulation controller generates the LCR interrupt (if enabled) when the PC writes these bits.

NOTE

The 16550 emulation logic will write a zero to the bit 8 location when the character length is programmed to 7-bits.

Set Break—Break Control Bit

When the PC sets this bit to one, the 16550 emulation controller will transfer the receive break indication to the 68000 core. It is done by incrementing the BRKEC counter, closing the receive buffer, setting the BR bit (if a buffer is currently open) and generating the BRK interrupt (if enabled). In addition, the 16550 emulation controller generates the LCR interrupt (if enabled). When the PC clears this bit, the 16550 emulation controller generates the LCR interrupt (if enabled). This enables the 68000 core to measure the BREAK length.

DLAB—Divisor Latch Access Bit

The PC must set this bit to one to access the divisor latches during a read or write operation. The PC must set this bit to zero to access the receiver buffer, the transmitter holding register, or the interrupt enable register.

4.4.4.1.2 Line Status Register (LSR) (Read Only).

LSR Base + $8F6

7	6	5	4	3	2	1	0
RX FIFO ERROR	TRANSMITTER EMPTY	TRANSMITTER HOLDING REGISTER	BREAK INTERRUPT	FRAMING ERROR	PARITY ERROR	OVERRUN ERROR	DATA READY
RESET							
0	O	0	0	0	0	0	0

Read-Only by the PC—68000 can set the Overrun bit

Data Ready

0 = The receiver data register or FIFO is empty
1 = A character has been transferred by the RISC controller into the receiver data register or FIFO.

Overrun

This bit is set by the 68000 core when the transmit buffer has not been transmitted for a long period of time. It is done to emulate the 16550 Rx FIFO overrun.

This bit is reset when the PC reads the LSR.

Parity Error

This bit is set by the 68000 core, by setting the P bit in the transmit buffer descriptor. This bit is reset when the PC reads the LSR. In FIFO mode, the parity error bit will be set when the character associated with the parity error moves to the top of the FIFO.

Framing Error

This bit is set by the 68000 core indirectly when the 68000 core sets the FR bit in the transmit buffer descriptor. This bit is reset when the PC reads the LSR register. In FIFO mode, the framing error bit is set when the character associated with the framing error moves to the top of the FIFO.

Break Interrupt (BI)

This bit is set by the 68000 core by issuing the STOP TRANSMIT command. This bit is reset when the PC reads the LSR register. In FIFO mode, this error bit will be set when the character associated with the error moves to the top of the FIFO. When break occurs, only one character is written into the FIFO.

NOTE

> Bits 1-4 are error bits that produce a receiver line status interrupt whenever one of these bits is set and the interrupt is enabled.

Transmitter Holding Register Empty (THRE)

0 = The transmitter holding register is written by the PC. In the FIFO mode at least one byte is written to the FIFO.
1 = The transmitter is ready to accept a new character. A character is transferred from the transmitter holding register to the transmitter shift register emulation. In the FIFO mode, the Tx FIFO is empty.

Transmitter Empty (TEMT)

0 = The transmitter holding register contains a character. In the FIFO mode at least one byte is written to the FIFO.
1 = The transmitter holding register and the transmitter shift register emulation are both empty. In the FIFO mode, the Tx FIFO is empty.

LSR7

In the NS16450 mode this bit is 0. In the FIFO mode, this bit is set when there is at least one parity error, framing error, or break indication in the FIFO. It is cleared when the PC reads the LSR, if there are no subsequent errors in the FIFO.

4.4.4.1.3 FIFO Control Register (FCR) (Write Only).

FCR — PC Access Only

	7	6	5	4	3	2	1	0
	RX FIFO TRIGGER		RESERVED	RESERVED	DMA MODE	XMITFIFORST	RCVRFIFORST	FIFO ENABLE
RESET	0	O	0	0	0	0	0	0

Write-Only—PC

FIFO Enable

0 = Clear all bytes in XMIT and RCVR FIFOs.
1 = Enables both FIFOs.

NOTE

> When changing from FIFO mode to NS16450 mode and vice versa, data is automatically cleared from the FIFOs. This bit must be one when other FCR bits are written.

Receiver FIFO Reset

When the PC writes a one to this bit, all bytes are cleared in the RCVR FIFO, and the RCVR FIFO counter is reset. The one that is written to this bit is cleared immediately (i.e. it is not latched).

Transmitter FIFO Reset

When the PC writes a one to this bit, all bytes in the XMIT FIFO are cleared and the XMIT FIFO counter is reset.The one that is written to this bit is cleared immediately (i.e. it is not latched).

DMA Mode Select

This bit is not implemented because the DMA mode is not supported.

Receiver FIFO trigger

These bits are used to set the trigger level for the receiver FIFO interrupt.

Table 4-13. Receiver FIFO Trigger Level (Bytes)

FCR7	**FCR6**	**RCVR FIFO Trigger Level (Bytes)**
0	0	01
0	1	04
1	0	08
1	1	14

4.4.4.1.4 Interrupt Identification Register (IIR) (Read Only). The interrupts are prioritized into four levels and recorded in the interrupt identification register (IIR). The four levels of interrupt conditions in order of priority are: receiver line status, received data ready, transmitter holding register empty, and MODEM status. When the PC accesses the IIR, the contents of the register and all pending interrupts are frozen. Any new interrupts will be recorded and updated after the current access is terminated.

IIR PC Access Only

	7	6	5	4	3	2	1	0
	FIFO ENABLES		RESERVED	RESERVED	INTERRUPT ID			INTERRUPT PENDING
RESET	0	O	0	0	0	0	0	0

Read-Only–PC

Interrupt Pending

0 = An interrupt is pending and the IIR contents may be used as a pointer to the appropriate interrupt routine.
1 = No interrupt is pending

Interrupt ID

These bits are used to identify the highest priority interrupt pending, as indicated in Table 4-14.

Bits 4 and 5—These bits are always 0.

Bits 6 and 7—These bits are set when FCR0=1. In the 16450 mode they are always 0.

Table 4-14. Interrupt Control Functions

FIFO Mode Only	Interrupt Identification Reg			Interrupt Set and Reset Functions			
Bit 3	Bit 2	Bit 1	Bit 0	Priority Level	Interrupt Type	Interrupt Source	Interrupt Reset Control
0	0	0	1	-	None	None	-
0	1	1	0	Highest	Rx line status	Overrun or parity error or framing error or break interrupt	Reading the line status register
0	1	0	0	Second	Rx data available	Rx Data available or trigger level reached	Reading the Rx buffer register or the FIFO drops below the trigger level.
1	1	0	0	Second	Character timeout indication	No characters have been removed or input to the Rx FIFO during the last 4 character times and there was at least 1 character in it during this time.	Reading the receiver buffer register.
0	0	1	0	Third	Tx holding register empty	Tx Holding register empty	Reading the IIR register (if source of interrupt) or writing into the transmitter holding register.
0	0	0	0	Fourth	MODEM status	CTS, DSR, ring indication, or CD	Reading the MODEM status register

4.4.4.1.5 Interrupt Enable Register (IER). PC Access Only

7	6	5	4	3	2	1	0
0	0	0	0	EDSSI	ELSI	ETBEI	ERBFI
RESET 0	O	0	0	0	0	0	0

Read/Write–PC

ERBFI—Enable Received Data Available Interrupt

This bit enables the received data available interrupt to the PC (and timeout interrupts in the FIFO mode), when set to one.

ETBEI—Enable Transmitter Holding Register Empty Interrupt

This bit enables the transmitter holding register empty interrupt to the PC, when set to one.

ELSI —Enable Receiver Line Status Interrupt

This bit enables the receiver line status interrupt to the PC, when set to one.

EDSSI—Enable MODEM Status Interrupt

This bit enables the MODEM status interrupt to the PC, when set to one.

Bits 4 to 7—These bits are always 0.

4.4.4.1.6 MODEM Control Register (MCR).

MCR PC Access Only

7	6	5	4	3	2	1	0
0	0	0	Loop	Out2	Out1	RTS	DTR
RESET 0	O	0	0	0	0	0	0

Read/Write–PC

DTR—Data Terminal Ready

When this bit is set, the DTR status bit in the 16550 status register is forced to 1. When this bit is reset, the DTR status bit in the 16550 status register is forced to 0. DTR status change interrupt may be generated. See 4.4.5.12 16550 Event Register for more details.

RTS—Request To Send

When this bit is set, the RTS status bit in the 16550 status register is forced to 1. When this bit is reset, the RTS status bit in the 16550 status registers forced to 0. RTS status change interrupt may be generated. See 4.4.5.12 16550 Event Register for more details.

Out 1

When this bit is set, the Out 1 status bit in the 16550 status register is forced to 1. When this bit is reset, the Out 1 status bit in the 16550 status register is forced to 0. An Out 1 status change interrupt may be generated. See 4.4.5.12 16550 Event Register for more details.

Out 2

When this bit is set, the Out 2 status bit in the 16550 status register is forced to 1. When this bit is reset, the Out 2 status bit in the 16550 status register is forced to 0. Out 2 status change interrupt may be generated. See 4.4.5.12 16550 Event Register for more details.

Loop—Loopback Mode

When this pin is set to one the 16550 emulation controller is forced into the local loopback diagnostic feature. The following occurs:

The four MODEM control outputs (DTR, RTS, OUT 1, OUT 2) are internally connected to the four MODEM control inputs (DSR, CTS, RI and DCD).

The 68000 core will receive the loop interrupt (See Loop—Loopback Mode on page 47) and will have to transfer data between the TxBDs and the RxBD to emulate the loopback feature. The buffer length should only be one byte.

NOTE

To emulate the local loopback diagnostic, the slowdown mechanism should be enabled. It should be enabled by the user, otherwise the loopback will be done too quickly.

Bits 5-7 —Always 0.

4.4.4.1.7 MODEM Status Register (MSR).

MSR Base + $8F3

7	6	5	4	3	2	1	0
DCD	RI	DSR	CTS	DDCD	TERI	DDSR	DCTS
RESET 0	O	0	0	0	0	0	0

Read-Only–PC

Write-68000

DCTS—Delta Clear to Send

This bit indicates that the CTS bit has changed state since the last time it was read by the PC. This bit is reset to logic 0 whenever the PC reads the MSR register.

DDSR—Delta Data Set Ready

This bit indicates that the DSR bit has changed state since the last time it was read by the PC. This bit is reset to logic 0 whenever the PC reads the MSR register.

TERI—Trailing Edge of Ring Indicator

This bit indicates that the RI bit has changed from high to low state. This bit is reset to logic 0 whenever the PC reads the MSR register.

DDCD—Delta Data Carrier Detect

This bit indicates that the DCD bit has changed state. This bit is reset to logic 0 whenever the PC reads the MSR register.

NOTE

Whenever bits 0, 1, 2 or 3 are set to one, a MODEM status interrupt is generated to the PC.

CTS—Clear To Send

This bit emulates the complement of the CTS input. The 68000 core writes this bit. If bit 4 (loop) of the MCR is set to 1, this bit is equivalent to RTS in the MCR.

DSR—Data Set Ready

This bit emulates the complement of the DSR input. The 68000 core writes this bit. If bit 4 (loop) of the MCR is set to 1, this bit is equivalent to DTR in the MCR.

RI—Ring Indicator

This bit emulates the complement of the RI input. The 68000 core writes this bit. If bit 4 (loop) of the MCR is set to 1, this bit is equivalent to OUT 1 in the MCR.

DCD—Data Carrier Detect

This bit emulates the complement of the DCD input. The 68000 core writes this bit. If bit 4 (loop) of the MCR is set to 1, this bit is equivalent to OUT 2 in the MCR.

4.4.4.1.8 Divisor Latch (LS) - DLL. This register contains the low byte of the baud rate generator divisor latch. When the PC writes this register, the DLL interrupt (if enabled) will be generated to the 68000 core. See 4.4.5.12 16550 Event Register for more details. The 68000 core can read this register value and program SCC2 BRG according to this value.

NOTE

The 16550 requires a baud rate multiplied by one clock instead of the baud rate multiplied by sixteen clock used in the standard 68302 SCC in UART mode (i.e. to generate a 16550 9600 baud clock at 16.67MHZ, the clock divider bits in the SCON register should be programmed to 1735 instead of 108). The 4 bit dividers for the TX and RX FIFOs should also be programmed to match the number of bits per character being transferred to slow down the PC interrupt rate to a level to emulate a serialized UART.

4.4.4.1.9 Divisor Latch (LM) - DLM. This register contains the high byte of the baud rate generator divisor latch. When the PC writes this register, the DLM interrupt (if enabled) will be generated to the 68000 core. See 4.4.5.12 16550 Event Register for more details. The 68000 core can read the register value at program SCC2 BRG.

4.4.4.1.10 Receive Buffer Register (RBR). When the 68000 core has data to deliver to the PC, it prepares a buffer descriptor with associated data buffer. The RISC controller will deliver the data buffer characters into the receive buffer register. An interrupt will be generated to the PC if the receive data available interrupt is enabled and the ready bit is set. Also, the 68000 core may set the FE and P error bits in the transmit buffer descriptor (TxBD) and the RISC will transfer them with the data to the FIFO or the line status register. To slow down the transfer rate to the PC and emulate the same rate as a serial 16550, the PC interrupt request rate may be slowed by programming the EMR RX divider bits (see 4.4.5.2 16550 Emulation Mode Register (EMR)).

4.4.4.1.11 Transmit Holding Register (THR). When the PC writes to the transmit holding register, the RISC controller receives a request from the 16550 emulation logic. The RISC controller, when servicing the request, will read the register and will transfer the data into the Rx data buffer. When the RISC reads the register, the transmit holding register empty flag is set. To slow down the transfer rate and emulate the same rate as the serial 16550, the RISC controller request may be delayed by programming the EMR TX divider bits (see 4.4.5.2 16550 Emulation Mode Register (EMR)).

4.4.4.1.12 Scratchpad Register (SCR). This 8-bit read/write register does not control the 16550 in any way. It is intended as a scratchpad register to be used by the PC to hold data temporarily.

4.4.5 68000 Programming Model

SCC2 can be configured as a 16550 emulation controller by programming its mode register (SCM) into UART16550 mode (MODE1-MODE0 = 11). BRG2 should be programmed to the appropriate rate after reading the 16550 emulation divisor latch. It uses the same data structure as the SCCs in the other modes. The 16550 data structure supports multi-buffer operation. The user can program the 16550 emulation controller to accept or reject control characters. If a control character is rejected, an interrupt may be generated. The 16550 emulation controller enables the user to transfer break sequences, parity and framing errors to the PC via the buffer descriptor table. An indication of the status of the 16550 emulation transmit FIFO (IDLE) is reported through the status register, and a maskable interrupt is generated upon a status change. In its simplest form, the 16550 can function in a character-oriented environment. Each character is transmitted with accompanied stop bits and parity (as configured by the user), and received into separate one byte buffers. Reception of each buffer may generate a maskable interrupt.

4.4.5.1 16550 MEMORY MAP. When the 16550 emulation controller is enabled, the 68PM302 overlays the structure illustrated in Table 4-2, with the 16550 emulation controller specific parameters, described in Table 4-15.

Table 4-15. 16550 Specific Parameter RAM

Address	Name	Width	Description
SCC2 Base+9C	**MAX_IDL**	Word	Maximum IDLE characters
SCC2 Base+9E	IDLC	Word	Temporary IDLE counter
SCC2 Base+A0	**BRKCR**	Word	Break count register (transmit)
SCC2 Base+A2	res	Word	-
SCC2 Base+A4	res	Word	-
SCC2 Base+A6	res	Word	-
SCC2 Base+A8	BRKEC	Word	Receive break condition counter
SCC2 Base+AA	res	Word	-
SCC2 Base+AC	res	Word	-
SCC2 Base+AE	**RCCR**	Word	Receive control character register
SCC2 Base+B0	**CHARCTER1**	Word	CONTROL character 1
SCC2 Base+B2	**CHARCTER2**	Word	CONTROL character 2
SCC2 Base+B4	**CHARCTER3**	Word	CONTROL character 3
SCC2 Base+B6	**CHARCTER4**	Word	CONTROL character 4
SCC2 Base+B8	**CHARCTER5**	Word	CONTROL character 5
SCC2 Base+BA	**CHARCTER6**	Word	CONTROL character 6

Table 4-15. 16550 Specific Parameter RAM

Address	Name	Width	Description
SCC2 Base+BC	**CHARCTER7**	Word	CONTROL character 7
SCC2 Base+BE	**CHARCTER8**	Word	CONTROL character 8

The items above in **bold face** should be initialized by the user.

MAX_IDL.

The 16550 emulation TX FIFO is polled by the RISC controller every programmable amount of time. When the Tx FIFO is empty, the RISC controller assumes reception of an IDLE character. Once a character of data is received, the RISC controller counts any idle characters received. If a MAX_IDL number of idle characters are received before the next data character is received, and idle timeout occurs, the buffer is closed. This, in turn, can produce an interrupt request to the 68000 core to receive the data from the buffer. Thus, MAX_IDL provides a convenient way to demarcate frames in the 16550 mode.

NOTE

Program MAX_IDL to $0001 for the minimum timeout value; program MAX_IDL to $0000 for the maximum timeout value.

IDLC

This value is used by the RISC to store the current idle emulation counter value in the MAX_IDL timeout process. IDLC is a down counter. It does not need to be initialized or accessed by the user.

BRKCR.

The 16550 emulation controller will send an a break character sequence towards the PC whenever a STOP TRANSMIT command is given. The Break Interrupt bit in the emulation line status register will be set. (See Line Status Register (LSR) (Read Only) on page 28 for more details). The data sent towards the 16550 emulation receiver is the data written in character8 in the control characters table. The number of break characters sent by the 16550 emulation controller is determined by the value in BRKCR.

RCCR,CHARACTER8-1.

These characters define the receive control characters, on which interrupts may be generated.

4.4.5.2 16550 EMULATION MODE REGISTER (EMR). This register programmed by the 68000 core is used to enable the RISC controller reception and transmission processes. In addition, this register is used to program the counters used for reception, to slow down transmissions, for timeout in the 16550 emulation Rx FIFO, and the Rx buffer descriptor IDLE count.

EMR Base + $8F0

15	14	13	12	11	10	9	8
RES	RES	NWAIT	FRZ	ENT	ENR	TX DIVIDER	
RESET				0			
0	0	0	0		0	0	0

0

7	6	5	4	3	2	1	0
TX DIVIDER		RX DIVIDER				TSDE	RSDE
RESET 0	0	0	0	0	0	0	0

Read/Write–68000

The 16550 controller uses the SCC2 interrupt event and mask registers and the SCC2 parameter RAM. The 16550 controller BRG uses the SCC2 SCON register to set the BRG clock frequency which can be further divided by the two 4-bit counters. A divide by two block can be enabled from either the system clock source or the TIN pin depending on where the baud rate clock is supplied.

NOTE

IMP clock to 16550 BRG rate should not exceed a ratio of 3:1.

NWAIT—PCMCIA No WAIT Cycle

The PC accesses the 16550 emulation registers through the PCMCIA interface on I/O space. These read and write cycles are synchronized with the IMP clock. The IMP clock rate affects the synchronization time and the cycle length. When using lower rate IMP clocks, $\overline{\text{WAIT}}$ should be asserted on the PCMCIA cycle to delay the completion of the cycle.

0 = $\overline{\text{WAIT}}$ will be asserted.
1 = $\overline{\text{WAIT}}$ will not be asserted. The 302 clock rate should be equal or higher than 25 Mhz, otherwise the data will not be read or written correctly.

FRZ—Freeze Transmission

This bit allows the user to halt the 68000 data transfer into the 16550 emulation receive FIFO. Data transfer will continue from the next character in the buffer when this bit is reset to zero.

0 = Normal operation (or resume transmission after FRZ is set)
1 = The RISC controller completes the current transfer into the 16550 emulation receive FIFO and then stops transferring data.

ENT—Enable Transmitter

0 = The RISC controller transmission process is disabled.
1 = The RISC controller transmission process is enabled

ENR—Enable Receiver

0 = The RISC controller reception process is disabled.
1 = The RISC controller reception process is enabled

Tx Divider

This 4-bit value is loaded into a 4-bit countdown counter. This counter divides BRG2 input clock. The 16550 emulation transmit FIFO will request the RISC controller to transfer data into memory when the counter reaches zero and the FIFO is not empty. This mechanism will slow down the transfer rate and emulate the PC interrupts rate as if the data transfer was done serially. When the FIFO is empty, the RISC controller will identify the request as an IDLE reception for use by the close BD mechanism (See 16550 Rx Buffer Descriptor (Rx BD) on page 42).

NOTE

In this case, the PC is transmitting to the 68000 core.

Rx Divider

This 4-bit value is loaded into a 4-bit countdown counter. This counter divides the BRG2 input clock. The 16550 emulation receive FIFO requests the RISC controller to poll its Tx BD. When the BD is not empty, it will transfer data from memory into the FIFO, when the counter reaches zero and the FIFO is not full. This mechanism slows down the transfer rate and emulates the PC interrupt rate as if the data transfer was done serially. This counter is also used for the 16550 Rx FIFO time-out interrupt emulation.

NOTE

In this case, the 68000 is transmitting to the PC core.

TSDE—Transmitter Slowdown Enable

When set, this bit enables the transmitter slow-down mechanism. When cleared, the transmit counter is used only for IDLE emulation and the transfer rate is as fast as the RISC controller and the PC can operate.

NOTE

In this case, the PC is transmitting to the 68000 core.

RSDE—Receiver Slowdown Enable

When set, this bit enables the receiver slow-down mechanism. When cleared, the receive counter is used only for FIFO time-out interrupt emulation and the transfer rate is as fast as the RISC controller and the PC can transfer. In this case, transfers will probably be limited by PC software, rather than the RISC and SDMA channels on the 68K bus. The IMP transmitter will poll the transmit BD ready bit every Tx counter time-out. When the buffer is ready, data will be transferred at the maximal rate.

NOTE

In this case, the 68000 is transmitting to the PC core.

4.4.5.3 16550 COMMAND SET. The following commands can be issued to the command register (CR).

4.4.5.4 16550 TRANSMIT COMMANDS.

4.4.5.4.1 *STOP TRANSMIT* Command . After a hardware or software reset, and after the channel has been enabled by setting the transmit enable bit in the EMR register, the channel will be in the transmit enable mode and will start polling the first buffer in the table every programmable number of clocks (as programmed by the Tx Divider bits in the EMR).

The channel *STOP TRANSMIT* command disables the transmission of characters to the 16550 emulation receive FIFO.

The 16550 transmitter will transmit a programmable number of break sequences to the 16550 emulation receive FIFO and then stop transmission. The number of break sequences (which may be zero) should be written to the break count register (BRKCR) before this command is issued to the 16550 Controller.

4.4.5.4.2 *RESTART TRANSMIT C*ommand. The channel *RESTART TRANSMIT* command enables the transmission of characters on the 16550 emulation receive FIFO. This command is expected by the 16550 controller after disabling the channel in the mode register (EMR), and after a STOP TRANSMIT command. The 16550 controller will resume transmission from the current transmitter buffer in the Tx BD Table.

4.4.5.5 16550 RECEIVE COMMANDS.

4.4.5.5.1 *ENTER HUNT MODE* Command. After a hardware or software reset, and after the channel has been enabled by setting the receive enable bit in the EMR register, the channel will be in the transmit enable mode and will use the first buffer in the table.

The *ENTER HUNT MODE* command is used to force the 16550 controller to close the current Rx BD (if data is being received using it). The command generates an RX interrupt (if enabled) as the buffer is closed. The 16550 controller will resume reception using the next BD once a single IDLE character is received.

4.4.5.6 16550 CONTROL CHARACTERS (RECEIVER). The 16550 controller has the capability to recognize special control characters transferred from the PC to the card. These characters may be used when the 16550 functions in a message oriented environment. Up to eight control characters may be defined by the user in the control characters table. Each of these characters may be either written to the receive buffer (upon which the buffer is closed and a new receive buffer taken) or rejected. If rejected, the character is written to the received control character register (RCCR) in internal RAM and a maskable interrupt is generated. This method is useful for notifying the user of the arrival of control characters (e.g. XOFF) that are not part of the received messages.

The 16550 uses a table of 16-bit entries to support control character recognition. Each entry consists of the control character, a valid bit, and a reject character bit. The control characters table is described in Table 4-16.

Table 4-16. 16550 Control Character Table

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET + 0									RCCR							
OFFSET + 2	E	R							CHARACTER1							
OFFSET + 4	E	R							CHARACTER2							
OFFSET + 6																
								•								
								•								
								•								
OFFSET + 10	E	R	REA	I	CT	0	0	0	CHARACTER8							

RCCR Received Control Character Register

Upon a control character match for which the Reject bit is set, the 16550 controller will write the control character into the RCCR and generate a maskable interrupt. The core must process the interrupt and read the RCCR before a second control character arrives. Failure to do so will result in the 16550 controller overwriting the first control character.

CHARACTER8-1 =T Control Character Values

These fields define control characters that should be compared to the incoming character.

E—End of table

0 = This entry is valid. The lower 8 bits will be checked against the incoming character.
1 = The entry is not valid. This must be the last entry in the control characters table.

NOTE

In tables with 8 control characters, this bit is always 0.

R—Reject character

0 = The character is not rejected but written into the receive buffer. The buffer is then closed and a new receive buffer is used if there is more data in the message. A maskable (I-bit in the receive BD) interrupt is generated.
1 = If this character is recognized it will not be written to the receive buffer. Instead, it is written to the received control characters register (RCCR) and a maskable interrupt is generated. The current buffer is not closed when a control character is received with R set.

4.4.5.6.1 Transmission of Out-of-Sequence Characters (Transmitter). Transmission of out-of-sequence characters to the PC is also supported by the 16550 controller, and is normally used for the transmission of flow control characters such as XON or XOFF. This is performed using the last (eight) entry in the control characters table. The 16550 controller will poll this character whenever the transmitter is enabled for 16550 controller operation. This includes during buffer transmission, and when no buffer is ready for transmission. The character is transmitted at a higher priority than the other characters in the transmit buffer (if any), but does not preempt characters already in the 16550 receiver emulation FIFO.

E—Empty

Must be one to use this entry as a flow control transmission character. To use this entry instead as a receive control characters entry, this E bit (and all other E bits in the table) should be zero.

R—Reject

Must be zero to use this entry as flow control transmission character. For receive control characters entry, it maintains its functionality as previously defined.

REA—Ready

This bit is set by the 68000 core when the character is ready for transmission and will remain one while the character is being transmitted. The CP clears this bit after transmission.

I—Interrupt

If set, the core will be interrupted when this character has been transmitted. (The TX bit will be set in the 16550 event register.)

CHARACTER8—Flow Control Character Value

This value contains the character to be transmitted. This value may be modified only while the REA bit is cleared.

4.4.5.7 BREAK SUPPORT (RECEIVER). The 16550 controller offers very flexible BREAK emulation support for the receiver. See 4.4.5.9.2 BREAK Sequence for more details.

4.4.5.8 SEND BREAK (TRANSMITTER). A break is an all-zeros character without a stop bit(s). A break emulation is transferred to the PC by issuing the STOP TRANSMIT command. The 16550 controller sends a programmable number of break characters according to the break count register (BRKCR), and then reverts to idle or sends data if the RESTART TRANSMIT command was given before completion.

The break characters do not pre-empt characters already in the 16550 receiver emulation FIFO.

4.4.5.9 16550 ERROR HANDLING . The 16550 controller reports character reception and transmission error conditions via the channel buffer descriptors, the error counters, and the 16550 event register. The modem interface lines can be monitored by the port C pins.

4.4.5.9.1 IDLE Sequence Receive. An IDLE is detected when one character consisting of all ones is received. The 16550 controller emulates IDLE reception. The 16550 emulation TX FIFO is polled by the RISC controller every programmable amount of time. When the Tx FIFO is empty, the RISC controller assumes reception of an IDLE character. When the 16550 is receiving data into a receive buffer, and an IDLE is received, the channel counts the number of consecutive IDLE characters received. If the count reaches the value programmed into MAX_IDL, the buffer is closed and an RX interrupt is generated. If no receive buffer is open, this event does not generate an interrupt or any status information. The internal idle counter (IDLC) is reset every time a character is received.

4.4.5.9.2 BREAK Sequence. The 16550 controller offers very flexible BREAK support for the receiver. When the PC sets the Set Break bit in the LCR register or a BREAK sequence is received, the 16550 controller increments the break error counter (BRKEC), and issues the break (BRK) event in the 16550 event register, which can generate an interrupt if enabled. If the 16550 controller was currently in the process of receiving characters when the BREAK was received, it will also close the receive buffer and set the BR bit in the Rx BD, and write the RX bit in the event register, which can generate an interrupt if enabled. A long break sequence only increments the counter once.

Error Counter

BRKEC - Break Error Counter

4.4.5.10 16550 RX BUFFER DESCRIPTOR (RX BD). The CP reports information concerning the received data on a per buffer basis via buffer descriptors. The CP closes the current buffer, generates a maskable interrupt, and starts to receive data to the next buffer due to these events:

1. Reception of a user-defined control character (when the Reject bit = 0 in the control character table entry).
2. Detection of the receive buffer being full.
3. Reception of a MAX_IDL number of consecutive IDLE characters.
4. Issuing the ENTER HUNT MODE command

The first word of the Rx BD contains control and status bits. Its format is detailed below.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET + 0	E	X	W	I	C	—	—	ID	—	—	BR	—	—	—	—	—
OFFSET + 2	DATA LENGTH															
OFFSET + 4	RX DATA BUFFER POINTER															
OFFSET +6																

E—Empty

0 = The data buffer associated with this BD has been filled with received data, or data reception has been aborted due to an error condition. The core is free to examine or write to any fields of this Rx BD. The CP will not use this BD again while the empty bit remains zero.

1 = The data buffer associated with this BD is empty, or reception is currently in progress. This Rx BD and its associated receive buffer are owned by the CP. Once the E bit is set, the 68000 core should not write any fields of this Rx BD.

X—External Buffer

0 = The buffer associated with this BD is in internal dual-port RAM.

1 = The buffer associated with this BD is in external memory.

W—Wrap (Final BD in Table)

0 = This is not the last buffer descriptor in the Rx BD table.
1 = This is the last buffer descriptor in the Rx BD table. After this buffer has been used, the CP will receive incoming data into the first BD in the table, allowing the user to use fewer than eight BDs to conserve internal RAM.

NOTE

> The user is required to set the wrap bit in one of the first eight BDs; otherwise, erratic behavior may occur.

I—Interrupt

0 = No interrupt is generated after this buffer has been filled.
1 = The RX bit in the 16550 event register will be set when this buffer has been completely filled by the CP, indicating the need for the 68000 core to process the buffer. The RX bit can cause an interrupt.

C—Control Character

0 = This buffer does not contain a control character.
1 = This buffer contains a control character. The last byte in the buffer is one of the user defined control characters.

ID—Buffer Closed on Reception of Idles

The buffer was closed due to the reception of the programmable number of consecutive IDLE sequences (defined in MAX_IDL).

BR—Break Received

A break sequence was received while receiving data into this buffer.

Data Length

Data length is the number of octets written by the CP into this BD's data buffer. It is written by the CP once as the BD is closed.

NOTE

> The actual amount of memory allocated for this buffer should be greater than or equal to the contents of the maximum receive buffer length register (MRBLR).

Rx Buffer Pointer

The receive buffer pointer, which always points to the first location of the associated data buffer, may be even or odd. The buffer may reside in either internal or external memory.

4.4.5.11 16550 TX BUFFER DESCRIPTOR (TX BD). Data is presented to the CP for transmission on the 16550 channel by arranging it in buffers referenced by the channel's transmit buffer descriptor table. The CP confirms transmission via the buffer descriptors to inform the processor that the buffers have been serviced. The first word contains status and control bits.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET + 0	R	X	W	I	ERR		0	—	—	—	—	—	—	—	—	—
OFFSET + 2	DATA LENGTH															
OFFSET + 4	TX DATA BUFFER POINTER															
OFFSET +6																

R—Ready

0 = The data buffer associated with this BD is not currently ready for transmission. The user is free to manipulate this BD or its associated data buffer. The CP clears this bit after the buffer has been transmitted or after an error condition is encountered.
1 = The data buffer, which has been prepared for transmission by the user, has not been transmitted or is currently being transmitted. No fields of this BD may be written by the user once this bit is set.

X—External Buffer

0 = The buffer associated with this BD is in internal dual-port RAM.
1 = The buffer associated with this BD is in external memory.

W—Wrap (Final BD in Table)

0 = This is not the last buffer descriptor in the Tx BD table.
1 = This is the last buffer descriptor in the Tx BD table. After this buffer has been used, the CP will transmit data from the first BD in the table, allowing the user to use fewer than eight BDs to conserve internal RAM.

NOTE

The user is required to set the wrap bit in one of the first eight BDs; otherwise, erratic behavior may occur.

I—Interrupt

0 = No interrupt is generated after this buffer has been serviced.
1 = The TX bit in the 16550 event register will be set when this buffer has been serviced by the CP, which can cause an interrupt.

Err—Parity and Framing Error

00 = No error is written into the 16550 emulation FIFO for the PC.
01 = Framing error is associated with the buffer's characters. The framing error will be set in the 16550 emulation line status register when the PC reads this character.
10 = When the transmit Slow Down is disabled (See TSDE—Transmitter Slowdown Enable on page 38) the next buffer transmission will be delayed by the counter time-out.
11 = A parity error is associated with the buffer's character. The parity error will be set in the 16550 emulation line status register when the PC reads this character.

NOTE

When parity error is transmitted to the PC, the buffer will only contain a single character.

NOTE

When error is transmitted to the PC, and the transmit Slow Down is disabled (See TSDE—Transmitter Slowdown Enable on page 38) the next buffer transmission will be delayed by the counter time-out.

Data Length

The data length is the number of octets that the CP should transmit from this BD's data buffer. It is never modified by the CP. This value should be normally greater than zero.

Tx Buffer Pointer

The Tx Buffer Pointer, which always points to the first location of the associated data buffer, may be even or odd. The buffer may reside in either internal or external memory.

4.4.5.12 16550 EVENT REGISTER. The 16550 event register (SCCE) is called the 16550 event register when the 16550 is enabled. It is a 16-bit register used to report events to the 68000 core recognized by the 16550 emulation controller and is also used generate interrupts. On recognition of an event, the 16550 controller will set the corresponding bit in the 16550 event register. Interrupts to the 68000 core generated by this register may be masked in the 16550 mask register.

The 16550 event register is a memory-mapped register that may be read by the 68000 at any time. A bit is cleared by writing a one (writing a zero does not affect a bit's value). More than one bit may be cleared at a time. All unmasked bits must be cleared before the CP will clear the internal interrupt request. This register is cleared at reset.

E550 Base + $898

15	14	13	12	11	10	9	8
RES	RES	RES	BRK	CCR	BSY	TX	RX
RESET 0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
IDL	LCR	LOOP	DL	OUT2	OUT1	RTS	DTR
RESET 0	0	0	0	0	0	0	0

Read/Write

Bits 15-13—Reserved. Should be written with zeros.

BRK—Break

This bit is set when a break character is received. This is the first break of a break sequence. Multiple break events will not be generated if a long break sequence is received.

CCR—Control Character Received

This bit is set when a control character was received (with reject character bit in the control character table (R bit = 1) and stored in the receive control character register (RCCR).

BSY—Busy Condition

This bit is set when a character was received and discarded due to lack of buffers. Reception continues as soon as an empty buffer is provided.

TX—Tx Buffer

This bit is set when a buffer has been transmitted over the 16550 channel. If CR = 1 in the Tx BD, this bit is set no sooner than when the last stop bit of the last character in the buffer begins to be transmitted. If CR = 0, this bit is set after the last character was written to the transmit FIFO.

RX—Rx Buffer

This bit is set when a buffer has been received over the 16550 channel. This event occurs no sooner than the middle of the first stop bit of the character that causes the buffer to be closed.

IDL—IDLE Sequence Status Changed

This bit is set when a change in the status of the serial line is detected on the 16550 channel. The real-time status of the line may be read in SCCS. Idle is entered when 16550 emulation FIFO has been empty for at least one full character time. It is exited when a character is received.

LCR—Line Control Register Write

This bit is set when the PC writes to the 16550 LCR emulation register.

DL—Divisor Latch

This bit is set when the PC writes to the 16550 divisor latch (either DLL or DLM) emulation registers

Loop—Loopback Mode

This bit is set when the PC changes the loop signal level by writing to the 16550 MODEM control emulation register. The status register may be read to determine loop current status.

Out2—Out2 Status Changed

This bit is set when the PC changes the Out2 signal level by writing to the 16550 MODEM control emulation register. The status register may be read to determine Out2 current status.

Out1—Out1 Status Changed

This bit is set when the PC changes the Out1 signal level by writing to the 16550 MODEM control emulation register. The status register may be read to determine Out1 current status.

RTS—RTS Status Changed

This bit is set when the PC changes the RTS signal level by writing to the 16550 MODEM control emulation register. The status register may be read to determine RTS current status.

DTR—DTR Status Changed

This bit is set when the PC changes the DTR signal level by writing to the 16550 MODEM control emulation register. The status register may be read to determine DTR current status.

4.4.5.13 16550 MASK REGISTER. The 16550 mask register is a 16-bit read-write register with the same bit formats as the 16550 event register. If a bit in the 16550 mask register is a one, the corresponding interrupt in the event register will be enabled. If the bit is zero, the corresponding interrupt in the event register will be masked. This register is cleared upon reset.

4.4.5.14 16550 STATUS REGISTER. The 16550 status register is a 16-bit read-only register which allows the user to monitor real-time status conditions set by the PC.

S550 Base + $8AC

15	14	13	12	11	10	9	8
RES	RES	RES	RES	RES	RES	RES	RES
RESET 0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
RES	RES	Loop	ID	OUT2	OUT1	RTS	DTR
RESET 0	0	0	0	0	0	0	0

Read/Write

Out2—Out1 Status

This bit reflects the current status of the Out2 bit in the 16550 MODEM control emulation register.

Out1—Out1 Status

This bit reflects the current status of the Out1 bit in the 16550 MODEM control emulation register.

RTS—RTS Status

This bit reflects the current status of the RTS bit in the 16550 MODEM control emulation register.

DTR—DTR Status

This bit reflects the current status of the DTR bit in the 16550 MODEM control emulation register.

ID—Idle Status

The ID bit is set when the 16550 emulation FIFO has been empty for at least one full character time.

0 = The line is not currently idle.
1 = The line is currently idle.

Loop—Loopback Mode

This bit reflects the current status of the Loop bit in the 16550 MODEM control emulation register.

4.5 SERIAL COMMUNICATION PORT (SCP)

The functionality of the SCP has not changed. For any additional information on parameters, registers, and functionality, please refer to the *MC68302 Users' Manual*.

4.5.1 SCP Programming Model

The SCP mode register consists of the upper eight bits of SPMODE. The SCP mode register, an internal read-write register that controls both the SCP operation mode and clock source, is cleared by reset.

15	14	13	12	11	10	9	8
STR	LOOP	CI	PM3	PM2	PM1	PM0	EN

4.5.2 SCP Transmit/Receive Buffer Descriptor

The transmit/receive BD contains the data to be transmitted (written by the M68000 core) and the received data (written by the SCP). The done (D) bit indicates that the received data is valid and is cleared by the SCP.

15	14	8	7	0
D	RESERVED		DATA	

4.6 SERIAL MANAGEMENT CONTROLLERS (SMCS)

The functionality of the SMCs has not changed. For any additional information on parameters, registers, and functionality, please refer to the *MC68302 Users' Manual*.

4.6.1 SMC Programming Model

The operating mode of both SMC ports is defined by SMC mode, which consists of the lower eight bits of SPMODE. As previously mentioned, the upper eight bits program the SCP.

7	6	5	4	3	2	1	0
—	SMD3	SMD2	SMD1	SMD0	LOOP	EN2	EN1

4.6.2 SMC Memory Structure and Buffers Descriptors

The CP uses several memory structures and memory-mapped registers to communicate with the M68000 core. All the structures detailed in the following paragraphs reside in the dual-port RAM of the IMP. The SMC buffer descriptors allow the user to define one data byte at a time for each transmit channel and receive one data byte at a time for each receive channel.

4.6.2.1 SMC1 RECEIVE BUFFER DESCRIPTOR. The CP reports information about the received byte using this (BD).

15	14	13	12	11	10	9	8	7	0
E	L	ER	MS	—		AB	EB	DATA	

4.6.2.2 SMC1 TRANSMIT BUFFER DESCRIPTOR. The CP reports information about this transmit byte through the BD.

15	14	13	12 10	9	8	7 0
R	L	AR	—	AB	EB	DATA

4.6.2.3 SMC2 RECEIVE BUFFER DESCRIPTOR. In the IDL mode, this BD is identical to the SMC1 receive BD. In the GCI mode, SMC2 is used to control the C/I channel.

15	14 6	5 2	1	0
E	RESERVED	C/I	0	0

4.6.2.4 SMC2 TRANSMIT BUFFER DESCRIPTOR. In the IDL mode, this BD is identical to the SMC1 transmit BD. In the GCI mode, SMC2 is used to control the C/I channel.

15	14 6	5 2	1	0
R	RESERVED	C/I	0	0

R—Ready

0 = This bit is cleared by the CP after transmission to indicate that the BD is now available to the M68000 core.

1 = This bit is set by the M68000 core to indicate that the data associated with this BD is ready for transmission.

Bits 14–6—Reserved for future use; should be set to zero by the user.

C/I—Command/Indication Channel Data

Bits 1–0—These bits should be written with zeros by the M68000 core.

SECTION 5
PCMCIA CONTROLLER

The MC68PM302 includes a PCMCIA interface, implemented including the registers and signal pins necessary to provide a fully functional PC Card slave interface. In addition, there is a mechanism which allows the PC to directly access resources on the 68000 bus.

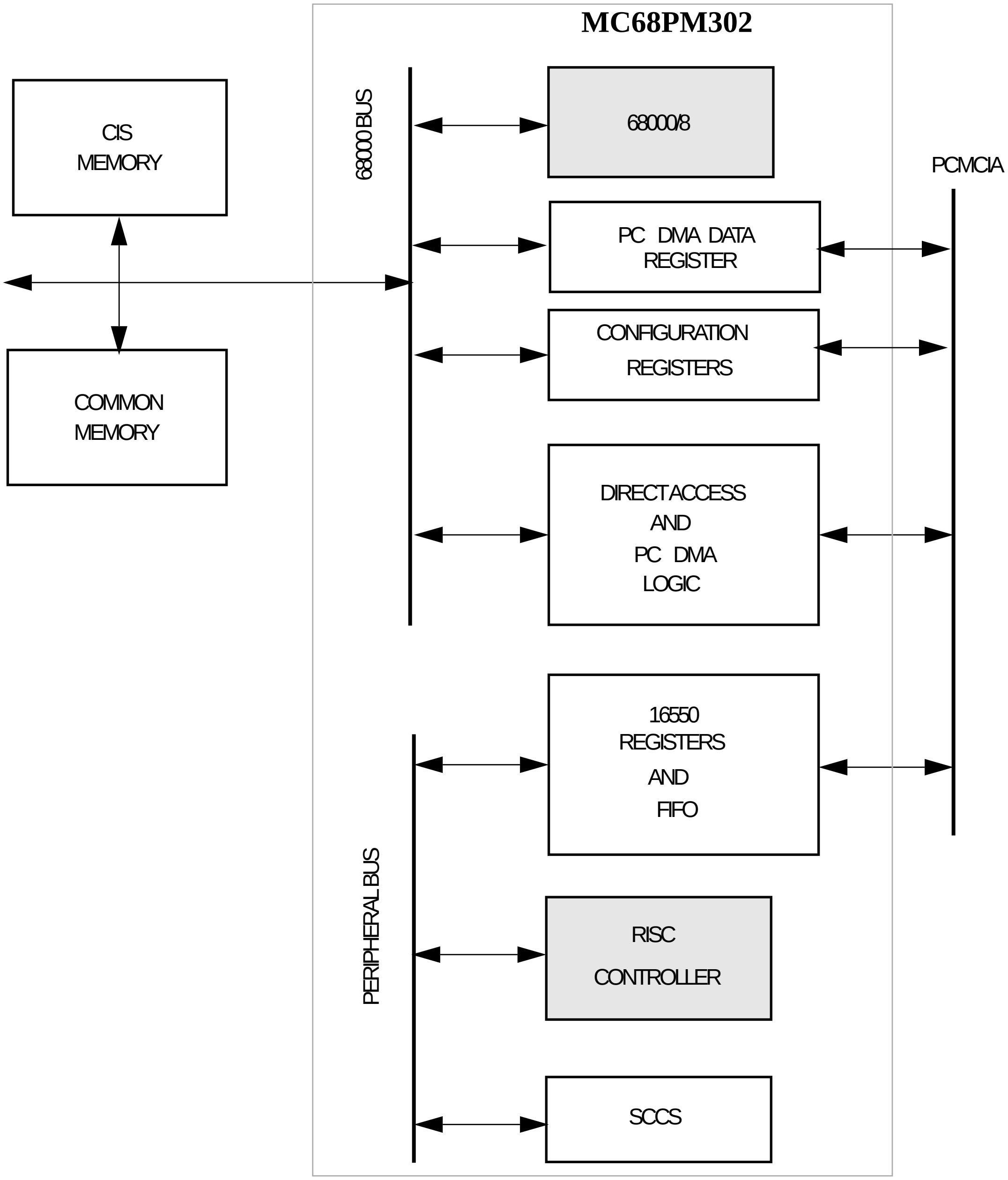


Figure 5-1. PCMCIA Architecture

The following are PCMCIA controller key features:

- Fully Supports PCMCIA Standard 2.1
- Supports Five Configuration Registers
  - —Configuration Option Register
  - —Card Configuration and Status Register
  - —Pin Replacement Register
  - —Socket and Copy Register
  - —I/O Event Indication Register
- Supports Attribute, Common Memory and I/O Space Access with Optional $\overline{\text{WAIT}}$ Handshake Mechanism
- Common Memory Spaces Are Mapped into 68000 Space Using either of Two Common Memory Base Address Registers
  - —Base Address Registers Are Programmable by either the PC or 68000 Core
  - —Base Register Can Be Used as Transmit and Receive Buffer Pointers for Ultra-Fast Communication Data Transfers
  - —Supports Cycle Steal Transfers and Burst Transfers Arbitration Scheme for Common Memory Accesses
- Supports Full 64 Megabytes of Attribute Memory Space Addressing
  - —Card Information Structure Mapped into 68000 Space with Special Base Address Register
- UART 16550 Registers and FIFO Mapped into I/O Space for x86 PC Compatibility
  - —I/O Pin Definitions Supported
- Supports Pull-Up/Pull-Down Resistors on PCMCIA Pins (PUCR Register)
- Supports Two Ring Indication Handling Methods:
  - —I/O Event Indication Register RI Enable and Event Bits
  - —Direct Connection to $\overline{\text{STSCHG}}$ Signal
- Supports PC power management of the card
- Internal Register and Chip Select Lockout of PCMCIA Accesses
- Supports PCMCIA DMA Cycles According to ‘DMA Change Pages Document 0035, release 004’
- ExCA Compatible

5.1 PCMCIA CONTROLLER FUNCTIONAL OVERVIEW

The PCMCIA controller fully supports PCMCIA standard 2.1. The block diagram of the MC68PM302 PCMCIA controller is shown in Figure 5-2. The PCMCIA controller interfaces between the PCMCIA bus, and both the 68000 bus and the UART16550 registers. Depending on the memory access mode, the controller will handle the transfer of data from or to the appropriate source or destination. Attribute memory accesses are supported for both the card configuration registers and the card information structure (CIS) memory. The CIS is located in external 68000 memory space. I/O accesses are relayed directly to the UART 16550 registers (Section 4 Communications Processor (CP)). Common memory accesses are mapped into 68000 memory. There is a unique common memory burst access mode which allows a much higher data transfer rate between the PC and the card because it eliminates the WAIT signal handshaking.

NOTE

> In this section, the PCMCIA master, or the computer to which the MC68PM302 slave interface will be plugged into and will communicate with, is referred to as the “PC”.

The PCMCIA controller also supports low power modes. The PC can place the card into one of the three supported STOP modes through the use of the PwrDwn bit in the CCSR. The PCMCIA controller is capable of waking up the MC68PM302 from the stand-by STOP mode when any access is made on the PCMCIA bus. Setting the power down bit can also cause a wake-up from any of the three stop modes.

The PCMCIA controller also supports the ring and packet indication for modem and other I/O cards. In addition to being able to wake up from low power modes when ring indicate (RI) is asserted, the MC68PM302 can optionally directly notify the PC without waking-up through a direct connection of $\overline{\text{RI}}$ to the $\overline{\text{STSCHG}}$ pin.

This chapter is structured to cover, first, the three memory space accesses specified by the PCMCIA committee: the attribute, I/O, and common memory spaces. The low power modes and wake up options are covered next, followed by the descriptions of all of the PCMCIA controller registers. Finally the PCMCIA card software initialization procedures are covered.

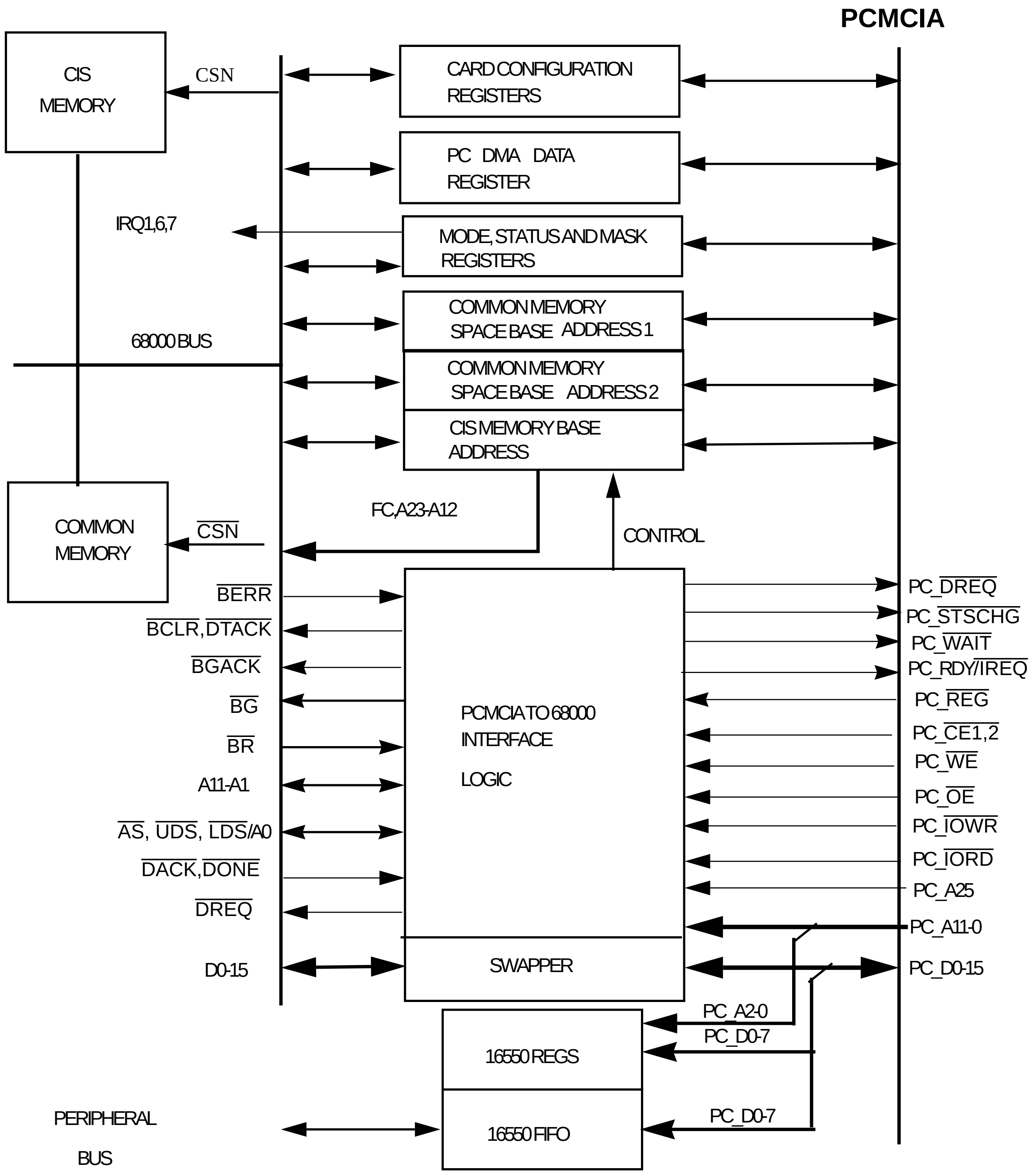


Figure 5-2. PCMCIA Controller Block Diagram

5.1.1 Attribute Memory Accesses

Attribute memory space accesses are used to access the configuration registers and the card information structure (CIS). Attribute memory accesses are generated when the PC asserts the PC_REG pin as shown in Table 5-1 and Table 5-2. As shown in Table 5-3, A25 selects between accessing the CIS with the CIS base address register (CISBAR), and direct asynchronous accesses to the card configuration registers and MC68PM302 PCMCIA controller registers. Attribute accesses can only be byte wide and memory locations and accesses must be at even addresses only. See Table 5-1, Table 5-2, and Table 5-3 for signal states and bus validity for the attribute memory read and write function.

5.1.2 Configuration Registers

The card configuration registers are built into the PCMCIA controller hardware. These registers, along with additional PCMCIA controller specific registers, are visible to the PCMCIA bus and have PCMCIA addresses in the attribute space as shown in Table 5-4. Accesses to these registers are asynchronous with respect to the system clock.

Table 5-1 Attribute Memory Read

Function Mode	PC_REG	PC_CE2	PC_CE1	PC_A0	PC_OE	PC_WE	PC_D15-PC_D8	PC_D7-PC_D0
Standby Mode	x	H	H	x	x	x	High-Z	High-Z
Byte Access	L L	H H	L L	L H	L L	H H	High-Z High-Z	Even- Byte Not Valid
Word Access	L	L	L	x	L	H	Not Valid	Even-Byte
Odd Byte Only Access	L	L	H	x	L	H	Not Valid	High-Z

Table 5-2 Attribute Memory Write

Function Mode	PC_REG	PC_CE2	PC_CE1	PC_A0	PC_OE	PC_WE	PC_D15-PC_D8	PC_D7-PC_D0
Standby Mode	x	H	H	x	x	x	xxx	High-Z
Byte Access	L L	H H	L L	L H	H H	L L	xxx xxx	Even- Byte xxx
Word Access	L	L	L	x	H	L	xxx	Even-Byte
Odd Byte Only Access	L	L	H	x	H	L	xxx	xxx

Table 5-3 Attribute Memory Space Map

PC_CE1	PC_REG	PC_OE	PC_WE	PC_A25	PC_A0	68000 Address	Selected Register or Space
L	L	L	H	L	L	CIS Base Reg (FC,A23-A11) || PCMCIA Address (A0-A10)	CIS Memory Read
L	L	H	L	L	L	CIS Base Reg (FC,A23-A11) || PCMCIA Address (A0-A10)	CIS Memory Write
L	L	L	H	H	L	PCMCIA Address (A0-A7)	Configuration Registers Read
L	L	H	L	H	L	PCMCIA Address (A0-A7)	Configuration Registers Write

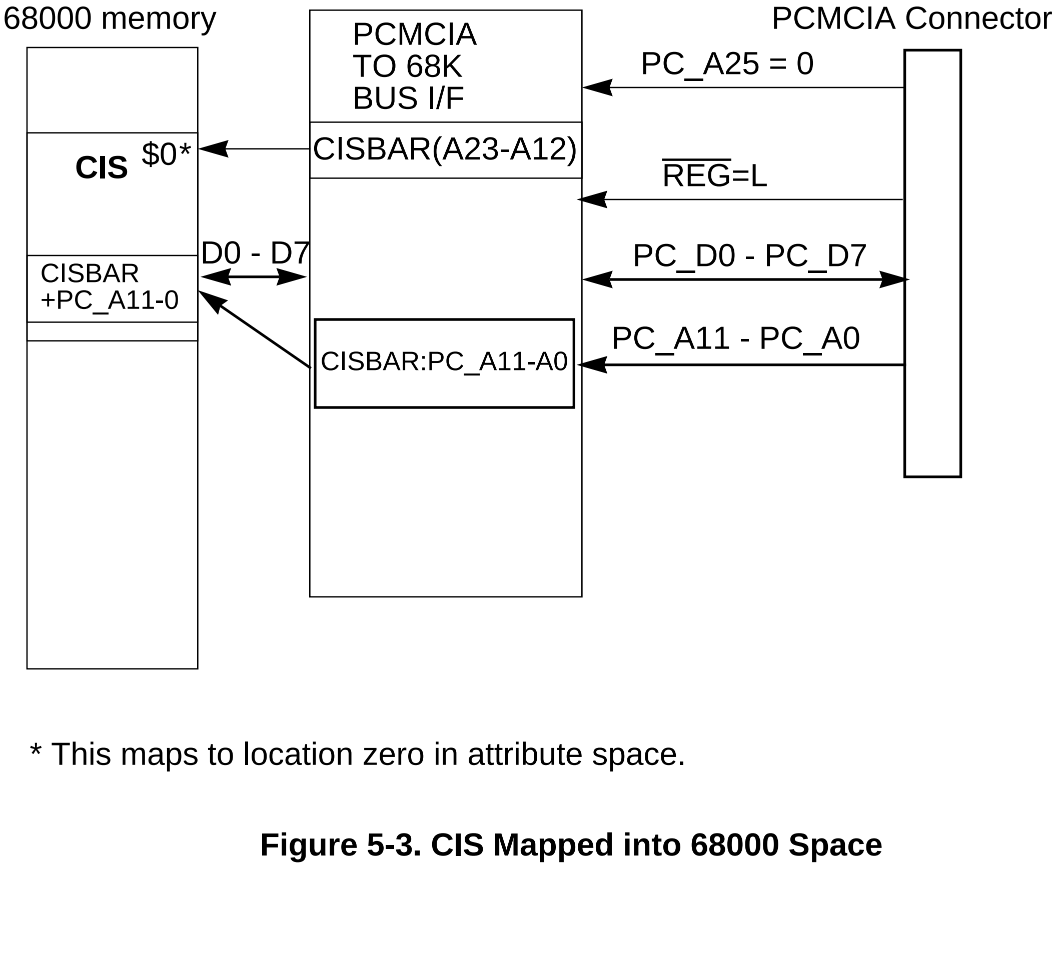
5.1.3 Card Information Structure

The card information structure is a data structure at address $0 of attribute space in the card that contains information about the card and its capabilities. The CIS will be located in 68000 bus memory space. Location $0 of the CIS is pointed to by CISBAR. Referring to Table 5-3, accesses to the CIS in attribute space are mapped into the 68000 address space with the concatenation of the CIS base address register (CISBAR=> A23-A12) and the low address bits taken from the PCMCIA address lines (PC_A11 - PC_A0). A25 must be equal to zero for CIS accesses to occur.

When the PC accesses the CIS by performing an attribute space read cycle to address 0 through $1FFFFFF (A25=0) the PCMCIA controller detects the cycle, asserts the $\overline{\text{WAIT}}$ signal on the PCMCIA lines and arbitrates for the 68000 bus. When granted, the controller generates the cycle on the 68000 bus. The high address bits and FC are taken from the CIS base address register, the low address bits are taken from the PCMCIA address lines. When the cycle is terminated with $\overline{\text{DTACK}}$, the 68000 bus will be released, data will be transferred to the PCMCIA data lines, and the $\overline{\text{WAIT}}$ signal will be negated.

NOTE

The PCMCIA attribute space accesses are only 8-bit. The external CIS memory must be at 68000 even addresses only



* This maps to location zero in attribute space.

Figure 5-3. CIS Mapped into 68000 Space

Table 5-4 PCMCIA Controller Registers Access Map

PC_CE1	PC_REG	PC_OE	PC_WE	PC_A25	Offset(PC_A7-PC_A0)	Selected Register or Space
L	L	L	H	H	00	Configuration Option Register
L	L	H	L	H	00	Configuration Option Register
L	L	L	H	H	02	Card Configuration and Status Register
L	L	H	L	H	02	Card Configuration and Status Register
L	L	L	H	H	04	Pin Replacement Register
L	L	H	L	H	04	Pin Replacement Register
L	L	L	H	H	06	Socket and Copy Register
L	L	H	L	H	06	Socket and Copy Register
L	L	L	H	H	08	Reserved1 (for future PCMCIA Definition)
L	L	H	L	H	08	Reserved1 (for future PCMCIA Definition)
L	L	L	H	H	0A	Reserved2 (for future PCMCIA Definition)
L	L	H	L	H	0A	Reserved2 (for future PCMCIA Definition)
L	L	L	H	H	0C	Reserved3 (for future PCMCIA Definition)
L	L	H	L	H	0C	Reserved3 (for future PCMCIA Definition)
L	L	L	H	H	0E	Reserved4 (for future PCMCIA Definition)
L	L	H	L	H	0E	Reserved4 (for future PCMCIA Definition)
L	L	L	H	H	20	PCMCIA Mode Register Read
L	L	H	L	H	20	PCMCIA Mode Register Write
L	L	L	H	H	2A	PCMCIA Event Register Read
L	L	H	L	H	2A	PCMCIA Event Register Write
L	L	L	H	H	30	CIS Base Address Register Read
L	L	H	L	H	30	CIS Base Address Register Write
L	L	L	H	H	38	Common Memory Space Base Address Register 1 Read
L	L	H	L	H	38	Common Memory Space Base Address Register 1 Write
L	L	L	H	H	40	Common Memory Space Base Address Register 2 Read
L	L	H	L	H	40	Common Memory Space Base Address Register 2 Write

5.1.4 I/O Space Accesses

The MC68PM302 fully emulates the 16550 UART (Section 4 - Communications Processor (CP)). The host can transfer data to the card by writing or reading the 16550 FIFO which is mapped into the PCMCIA address space. I/O accesses are initiated by the assertion of either the PC_IORD or the PC_IOWR signal and the PC_REG signal. All I/O accesses are connected directly to the 16550 emulation logic registers without using the 68000 bus.

The 16550 emulation registers addresses are located at addresses $0 through $7 in the PCMCIA I/O address space, therefore PCMCIA address lines 3 to 25 are ignored for I/O accesses. (See Table 5-7). Refer to Table 5-5 and Table 5-6 for signal states and bus validity for the I/O read and write function.

Table 5-5 I/O Input Accesses

Function Mode	PC_REG	PC_CE2	PC_CE1	PC_A0	PC_IORD	PC_IOWR	PC_D15-PC_D8	PC_D7-PC_D0
Standby Mode	x	H	H	x	x	x	High-Z	High-Z
Byte Access	L L	H H	L L	L H	L L	H H	High-Z High-Z	Even- Byte Odd-Byte
Word Access	L	L	L	x	L	H	Odd-Byte	Even-Byte
I/O inhibit (during DMA)	H	X	X	X	L	H	High-Z	High-Z
High Byte Only	L	L	H	x	L	H	Odd-Byte	High-Z

Table 5-6 I/O Output Accesses

Function Mode	PC_REG	PC_CE2	PC_CE1	PC_A0	PC_IORD	PC_IOWR	PC_D15-PC_D8	PC_D7-PC_D0
Standby Mode	x	H	H	x	x	x	xxx	xxx
Byte Access	L L	H H	L L	L H	H H	L L	xxx xxx	Even- Byte Odd-Byte
Word Access	L	L	L	x	H	L	Odd-Byte	Even-Byte
I/O Inhibit (during DMA)	H	X	X	X	H	L	xxx	xxx
High Byte Only	L	L	H	x	H	L	Odd-Byte	XXX

.

Table 5-7 Card I/O Space Address Map

PC_REG	PC_CE1	PC_CE2	PC_IORD	PC_IOWR	PC_A0 - PC_A2	Selected Register or Space
L	L	H	L	H	0-7*	16550 - 8 bytes read
L	L	H	H	L	0-7*	16550 - 8 bytes write

*See Table 2-6 for specific 16550 register address locations.

5.1.5 Common Memory and Direct Access Mode Accesses

Common memory reads and writes are initiated by the PC when it does a normal access by *not* asserting the PC_$\overline{\text{IORD}}$, PC_$\overline{\text{IOWR}}$, or the PC_$\overline{\text{REG}}$ pins. See Table 5-8 and Table 5-9 for signal states for reads and writes.

Common memory reads and writes are mapped directly onto the 68000 bus using a concatenation of the lower 12 address lines from the PCMCIA address bus and upper address lines from either of the two common memory base address registers (CMBAR1,2). The base address register to be used is selected by A25 (CMBAR1 is used when A25=1, and CMBAR2 is used when A25=0). Both the PC and the 68000 can program the CMBARs. See Table 5-10.

In addition to the being able to perform single common memory accesses between the PCMCIA bus and the 68000 bus there is also a feature that allows the PCMCIA controller to hold onto the 68000 bus and perform multiple reads or writes without having to rearbitrate for the 68000 bus. This allows the PCMCIA side access cycles to be carried out without having to assert the $\overline{\text{WAIT}}$ signal for each access. This mechanism works as follows:

When the PC initiates a common memory space cycle on the PCMCIA interface, the PCMCIA controller will assert the $\overline{\text{WAIT}}$ signal on the PCMCIA lines and arbitrate for the 68000 bus. When granted, the PCMCIA controller will generate the cycle on the 68000 bus. When the cycle is terminated with $\overline{\text{DTACK}}$, the PCMCIA controller may be programmed to maintain the bus ownership for the next cycle or release the bus. For burst access cycles (FAST=1 in PCMR register) bus ownership will be retained. Data is transferred to the PCMCIA data lines and the $\overline{\text{WAIT}}$ signal will be negated and the next cycle can take place without $\overline{\text{WAIT}}$ being reasserted. Subsequent accesses can occur until the PC has completed the burst and the PCMCIA bus is idle for a programmable number of 68000 bus clocks (CLKO) as specified by the ArbIDL bit in the PCMR.

The PCMCIA controller may be programmed to automatically increment the CMBAR when the PC reaches the current 4K page boundary defined by CMBAR (i.e. A11-A0=FFF). This mechanism is useful when the PC is using a DMA cycle to transfer data to or from the card because it eliminates the need to reprogram the base address register each time a page boundary is reached.

The two common memory base address registers (CMBARs) can be used to read and write to separate spaces in 68K memory, forming pointers to transmit and receive buffers. This can be especially useful in high speed communication cards where it is possible to bypass the use of the UART 16550 and simply transfer data directly to and from 68000 memory. Figure 5-4 shows how the CMBARs can be used as transmit and receive data buffer pointers. Once the data transfer begins, the PC can burst transfer data into 68K memory, incrementing the lower addresses for each transfer and using the A25 line to select between transmit and receive buffers. The CMBAR auto increment feature is also useful in this case if the buffer size needs to be larger than 4 Kbytes. By using this method, data transfer rates are limited only by the 68000 bus cycle rate or the PC bus transfer rate.

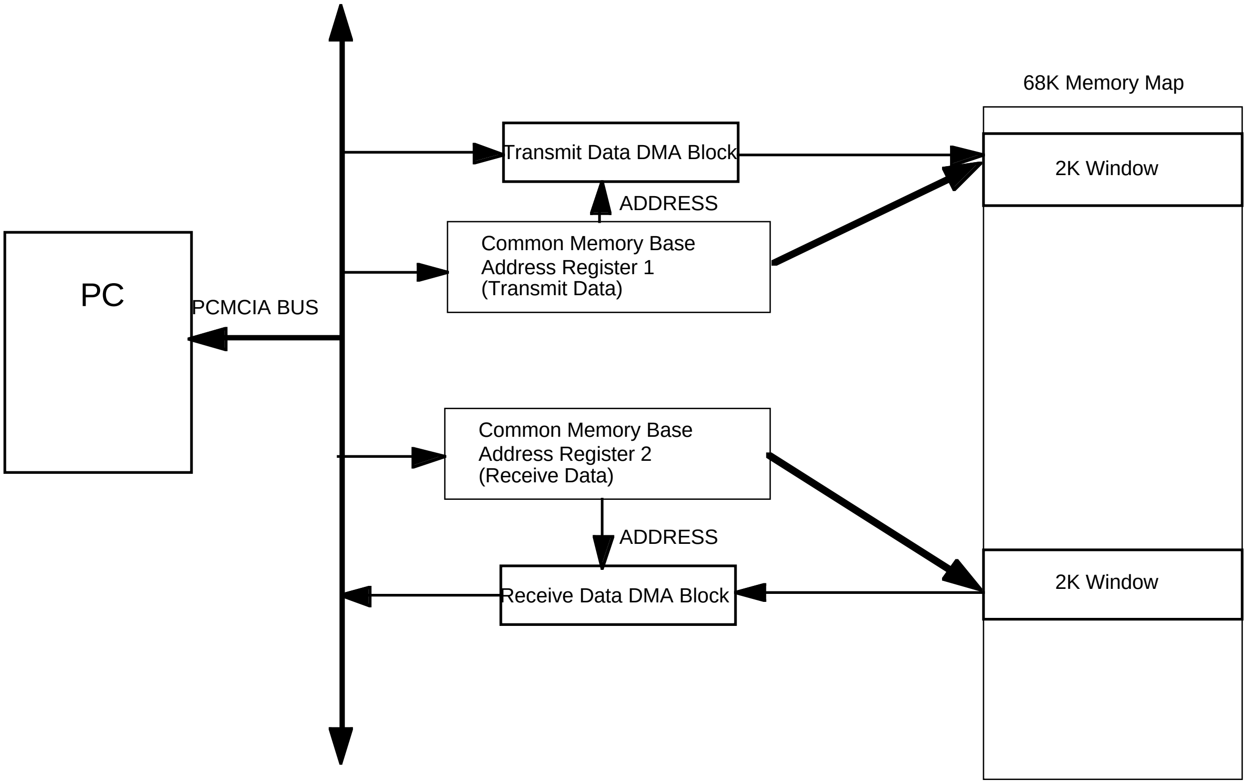


Figure 5-4. Use of the CMBARs for Fast Mode Transmit and Receive Data Transfers.

To eliminate paging entirely, the PCMCIA controller may be programmed not to drive the high address lines from the base register, but rather drive the ABUF pin which has the appropriate timing to control external address buffers. This address buffer control line is connected to external address buffer circuitry that interfaces the upper PCMCIA address bus lines to the 68000 bus. This feature is enabled by setting the ABUF bit in the PCMR register (See Figure 5-5)

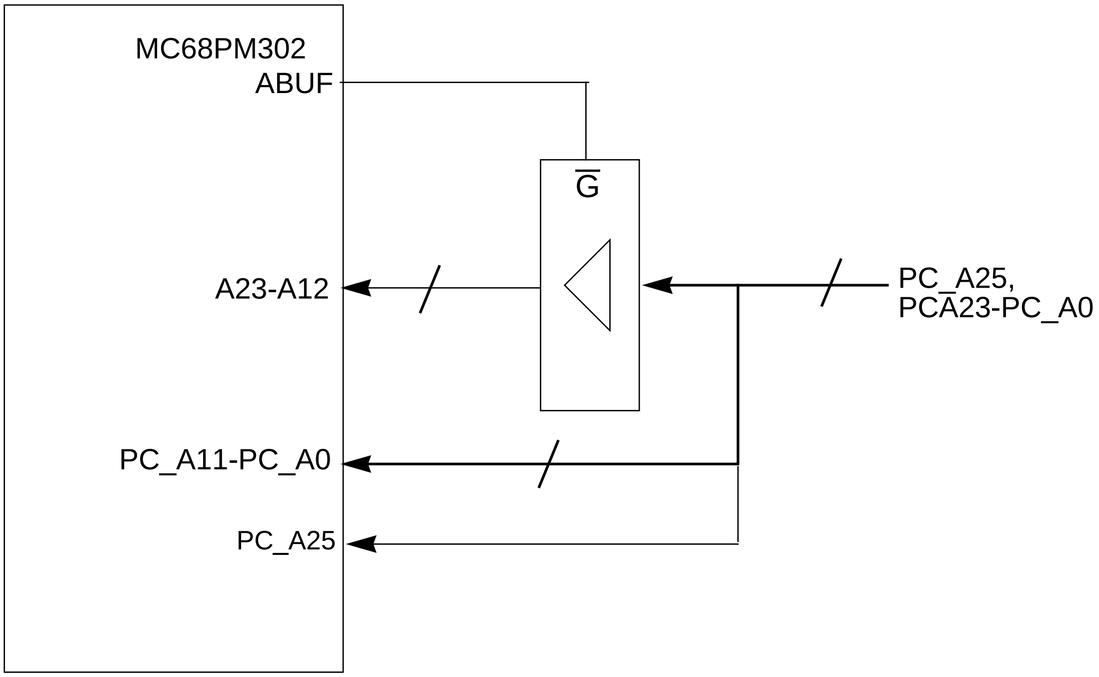


Figure 5-5. PCMCIA Direct Addressing Using ABUF

To summarize, common memory accesses can occur using two different modes:

- Normal mode (FAST =0): The PCMCIA controller will assert the $\overline{\text{WAIT}}$ signal on the PCMCIA lines and generate a 68000 cycle. When the cycle is terminated with $\overline{\text{DTACK}}$, data will be transferred to the PCMCIA data lines and the $\overline{\text{WAIT}}$ signal will be negated. The next normal mode cycle can then be initiated.
- Fast mode(FAST=1): This mode is used to eliminate the $\overline{\text{WAIT}}$ on the PCMCIA interface and perform fast direct access transfers. The first cycle (with arbitration) is as described above. From the second cycle until the burst is terminated, the PCMCIA controller **will not** assert the $\overline{\text{WAIT}}$ signal on the PCMCIA line. The cycle on the 68000 bus will use the memory interface signals $\overline{\text{WE}}$ and $\overline{\text{OE}}$ but not $\overline{\text{DS}}$. The cycle will be terminated when the PCMCIA cycle is terminated ($\overline{\text{DTACK}}$ and $\overline{\text{BERR}}$ are ignored in this mode). The PCMCIA controller will release the 68000 bus when the (PC) does not access the card for a programmable number of clocks.

NOTE

FAST mode accesses are not allowed for PC accesses to the IMP internal space. Erratic operation will result.

Table 5-8 Common Memory Read

Function Mode	PC_REG	PC_CE2	PC_CE1	PC_A0	PC_OE	PC_WE	PC_D15-PC_D8	PC_D7-PC_D0
Standby Mode	x	H	H	x	x	x	High-Z	High-Z
Byte Access	H H	H H	L L	L H	L L	H H	High-Z High-Z	Even- Byte Odd-Byte
Word Access	H	L	L	x	L	H	Odd-Byte	Even-Byte
Odd Byte Only Access	H	L	H	x	L	H	Odd-Byte	High-Z

Table 5-9 Common Memory Write

Function Mode	PC_REG	PC_CE2	PC_CE1	PC_A0	PC_OE	PC_WE	PC_D15-PC_D8	PC_D7-PC_D0
Standby Mode	x	H	H	x	x	x	xxx	xxx
Byte Access	H H	H H	L L	L H	H H	L L	xxx xxx	Even- Byte Odd-Byte
Word Access	H	L	L	x	H	L	Odd-Byte	Even-Byte
Odd Byte Only Access	H	L	H	x	H	L	Odd-Byte	XXX

Table 5-10 Card Common Memory Space Address Map

PC_REG	PC_CE1	PC_CE2	PC_OE	PC_WE	PCMCIA Offset Address	68000 Base Address	Selected Register or Space
H	L	H/L*	L	H	A25=1;	Common memory base reg1 FC,A23-11	Common memory read accesses into 68000 bus
H	L	H/L*	H	L	A25=1;	Common memory base reg1 FC,A23-11	Common memory write accesses into 68000 bus
H	L	H/L*	L	H	A25=0;	Common memory base reg2 FC,A23-11	Common memory read accesses into 68000 bus
H	L	H/L*	H	L	A25=0;	Common memory base reg2 FC,A23-11	Common memory write accesses into 68000 bus

*PC_CE2 will remain unasserted during byte accesses.

5.1.6 Protecting Memory and Internal Space from PCMCIA Accesses

Each chip select can be set to block PCMCIA accesses to its memory space. When the chip select protection bit is set, the external PCMCIA host cannot access the memory block. The protection for each chip select block can be enabled by setting the PCS(3-0) bits in the PCMCIA protection register (PPR). In addition, internal IMP registers and dual port RAM access from the PCMCIA bus can also be disabled using the PIR bit in the PPR. For more information see 3.8.1.3 PCMCIA Protection Register (PPR).

5.1.7 PCMCIA Controller Initialization

After hardware reset the PCMCIA controller assumes the default “memory only” card configuration, with the RDY/$\overline{\text{BSY}}$ signal being low to indicate the busy condition. The 68000 core should initialize the CIS base address register with the high address of the external CIS memory address. The CIS memory may be located in external IMP RAM/ROM which is pointed to by the CIS base address register (CISBAR). The MC68PM302 has five configuration registers, according to the PCMCIA standard, plus the three registers reserved for future use. These registers are located on the PCMCIA attribute address space and can be accessed by the PC when A25 is high (PCMCIA attribute addresses $2000000 or higher). The CIS’s configuration tuple TPCC_RADR should have the base address of these registers (PCMCIA attribute addresses $2000000 or higher).

The 68000 core software may then signal that the card is ready by setting the RDY/$\overline{\text{BSY}}$ line in the pin replacement register (PRR).

The host reads the CIS from attribute memory. This is done by the host performing an attribute space read cycle to address 0. The PCMCIA controller detects the cycle, asserts the $\overline{\text{WAIT}}$ signal on the PCMCIA lines and arbitrates for the 68000 bus. When granted, the controller generates the cycle on the 68000 bus. The high address bits and FC are taken from the CIS base address register, the low address bits are taken from the PCMCIA address lines. When the cycle is terminated with $\overline{\text{DTACK}}$, the 68000 bus will be released, data will be transferred to the PCMCIA data lines, and the $\overline{\text{WAIT}}$ signal will be negated. The host reads the CIS by this mechanism.

The host can then initialize the configuration registers. The host can access these registers directly without using the 68000 bus. When A25 is high on an attribute space cycle, the PCMCIA controller can asynchronously access the configuration registers. Those registers can also be accessed by the 68000 core. The 68000 core can read and write these registers like any other register in the 68000 address space. The PCMCIA controller may also generate an interrupt to the 68000 core when one of the registers is written by the PC.

After the initialization is completed, the MC68356 is configured for either PCMCIA common memory or I/O transfers depending on the value written in the configuration index of the configuration option register (COR). If the configuration index is non-zero, the card will respond to I/O cycles and the I/O pin functionality replaces the memory -card-only pin configuration. I/O data transfers are used to read and write the 16550 registers.

5.1.8 PCMCIA to 68000 Bus Access and Monitoring Options

The PCMCIA controller can be programmed to allow the SDMA, the core interrupt acknowledge routine, and external peripherals to use the 68000 bus between PCMCIA transfers by setting the BCLR bit in the PCMR register. The PCMCIA controller can also be programmed to generate a bus clear signal and gain mastership of the bus when it is not immediately granted upon request. This feature is enabled by setting the BCLROE bit in the PCMR.

If a PCMCIA to 68000 bus cycle is terminated with $\overline{\text{BERR}}$, the PCMCIA controller can be programmed to assert an interrupt to the PC through the INTRL bit in the PCMR. The $\overline{\text{BERR}}$ status bit can be read by the PC through the BERR bit in the PCHER register.

There is also a PCMCIA bus watchdog timer which can issue an interrupt to the PC as well as the 68000 core. The counter will count the $\overline{\text{WAIT}}$ length and, if expired, the PCMCIA cycle will be terminated and the PTIE interrupt to the PC will be generated (if enabled). The interrupt will be generated to the PC through the PTIE bit in the PCHER if the TIEn bit is set in the PCMR. The five-bit counter-preset is programmed through the TValue bits in the PCMR. The prescaler value is programmed in the TPres bits in the PCMR. The PCMCIA bus watchdog timer signal source is the IMP system clock.

NOTE

The features described in the above three paragraphs also apply for CIS accesses in attribute space.

NOTE

All PCMCIA locations (registers, CIS memory, etc.) that can be accessed by the host may also be accessed by the 68000 core. Power Down Options

The PC can place the MC68PM302 into low power mode by setting the power down (PwrDwn) bit in the CCSR. When the PC sets the PwrDwn bit, the PCMCIA controller will generate an interrupt to the 68000 core (if enabled through the INTRL bits in the PCMR). The IMP interrupt routine should place the devices on the card in a low power mode and then place itself into one of the low power modes listed in Table 2-3.

NOTE

The host should not change the PwrDwn bit in the CCSR while the card is busy. (i.e. The RDY/$\overline{\text{Bsy}}$ bit in the PRR is zero).

If the 68000 core is in a stand-by mode when the PwrDwn bit is set, the 68000 core will come out of stand-by mode to process the PwrDwn interrupt as above. Setting the PwrDwn bit will reset the RDY/$\overline{\text{Bsy}}$ bit in the pin replacement register (PRR) to zero. This bit will be set to one (RDY) when the IMP completes its transition to the power-down (STOP) mode.

Of the three different power down modes, the STAND_BY mode is the only one in which a normal host PCMCIA interface access will cause the IMP to resume operation (see 5.1.10.1 Wake Up on PCMCIA Access in STAND-BY mode).

The 68000 core can also place the card into one of four different low power modes when the card is idle for long periods of time and it is desired to conserve power (Table 2-3) even if the PC does not initiate the power down transition.

The PC can access the configuration registers while in power-down.

Table 5-11 IMP Low Power Modes

Low Power Mode	Method of Entry	Comments
Slow Go	Write to DF3 -DF0 in IOMCR	PCMCIA I/F functional, no wake up time, higher power consumption than other STOP mode.
Stand-by	STOP, LPM = 01	The imp can detect any PCMCIA access and wake-up. 2-5 cycle wake-up time.
Doze	STOP, LPM = 10	The IMP wakes up on setting of PwrDwn bit in CCSR, 2500 cycle wake-up time.
Stop	STOP, LPM = 11	The IMP wakes up on setting of PwrDwn bit in CCSR, 70000 max. cycle wake-up time.

In addition to placing itself in a STOP mode, the 68000 core can also lower the system clock frequency to save power by writing a higher value divide factor to the PLL clock divider. In this mode, the IMP and PCMCIA interface is fully functional. For more information see 2.4.4.2 Low Power Support.

5.1.9 PCMCIA Ring Indication

The MC68PM302 PCMCIA interface has several options for handling the ring indicate $\overline{\text{RI}}$ (or PB9) signal either by automatically passing the indication to the PC or by first allowing the 68000 core to process it. Figure 5-6 shows a diagram of possible routings of the ring indicate signal. Note that the IMP can be programmed to wake up from any of the three stop modes upon assertion of the RI signal.

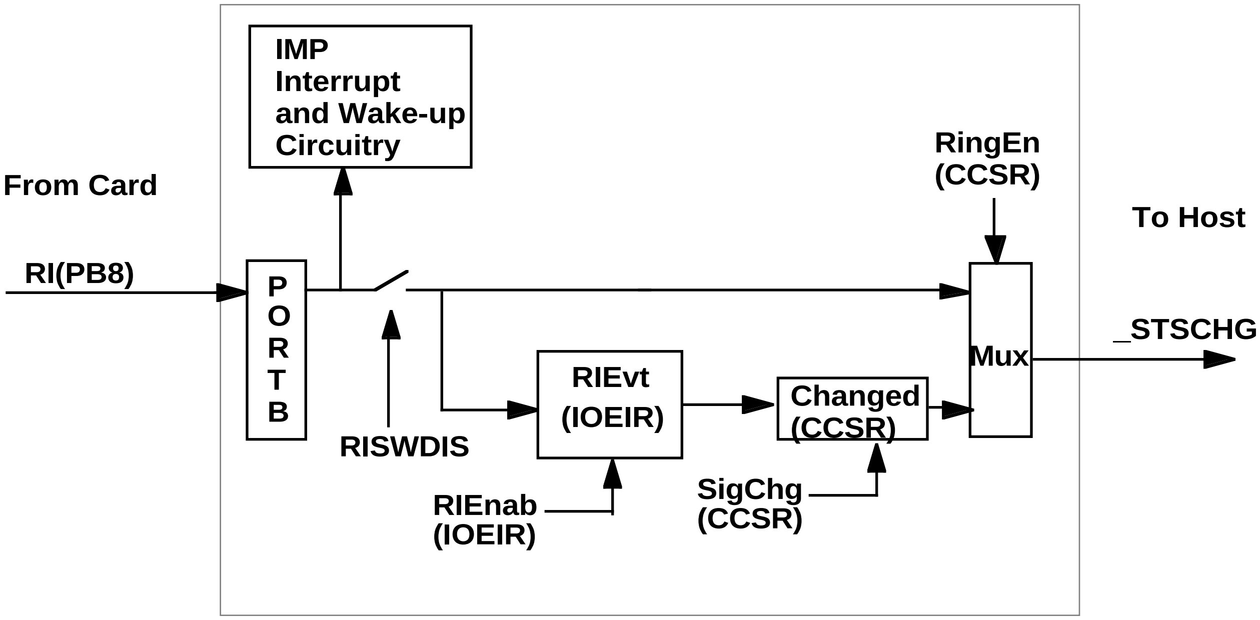


Figure 5-6. Ring Indicate (PB9) Connection Options

The MC68PM302 can transfer the $\overline{\text{RI}}$ (PB9) to the host using either of three programmable methods:

1. Pass the $\overline{\text{RI}}$ signal directly to the PC through the RIEvt bit in the I/O event indication register (IOEIR) which also will set the changed bit in the card configuration and status

register (CCSR) and can also optionally assert the STSCHG pin (if the SigChg bit is set in the CCSR). When the RI (PB9) is asserted, the RIEvt bit in the I/O event indication register(IOEIR) will be set to one. If the RIEnab has been set by the PC, the changed bit in the CCSR will be set. If the SigChg bit in the CCSR has been set to one also, the STSCHG pin will be driven low. In addition, the PC may poll the RIEvt bit in the IOEIR register, or the changed bit in the CCSR at all times. When the host detects the RI, it should clear out the RIEvt bit.

2. The 68000 may poll or take an interrupt off of the RI in the port B data register (PB9) and update the RIEvt bit in the IOEIR. This allows the 68000 software to debounce or otherwise process the RI signal to detect a valid ring signal before passing it on to the PC. The RISWDIS bit in the PCMR register should be set which will break the direct connection of the RI signal to the RIEvt bit in the IOEIR register (see Figure 5-6).
3. Connect the RI directly to the STSCHG pin by setting RingEn bit in the CCSR. This mode complies with Intel ExCA specification for host RI using STSCHG. When the RingEn bit in the CCSR register is set to one by the PC, the STSCHG line reflects the state of the RI (PB9) pin, all other status change events are disabled. When the RingEn bit is zero this function is disabled.

5.1.9.1 Wake Up Using the PwrDwn Bit

The host can wake up the card from any of the three STOP modes by resetting the PwrDwn bit in the CCSR. The IMP system clocks will start up, and the 68000 core will receive an interrupt. Resetting the PwrDwn bit will set the RDY/Bsy bit to zero in the pin replacement register. The 68000 core should set the RDY/Bsy bit to one upon completing its wake-up routine.

NOTE

> The user should reset enable and event bits in the IMP wake-up control register – IWUCR after it has recovered from STOP mode. (See 2.4.4.2.3 IMP Wake-Up from Low Power STOP Modes.)

5.1.10 Wake Up Options

There are several methods that allow the PC to wake up the IMP. The IMP will wake-up when:

1. The PC accesses the card and the IMP is in STAND-by mode.
2. The PwrDwn bit is reset.
3. The modem RI(PB9) or the PB10 signal is asserted, if enabled (See 2.4.4.2.3 IMP Wake-Up from Low Power STOP Modes.)
4. The (PIT) timer is enabled and expires, if enabled (See 2.4.4.2.3 IMP Wake-Up from Low Power STOP Modes.)
5. The IMP RESET and HALT is asserted.

5.1.10.1 Wake Up on PCMCIA Access in STAND-BY mode

The IMP will wake up on any of five different types of accesses to the PCMCIA interface if it is in the STAND-BY low power mode. The PCMCIA write event register (PCAWER) indicates the type of access that has taken place to wake-up the IMP. The PCMCIA write event mask registers (PCAWMR) is used to enable the wake-up mechanism for a particular type of PCMCIA access. The following accesses can be programmed as described above to wake up the IMP from the STANDY-BY mode:

- I/O space access.
- Attribute space Access
- Common memory space access (1 or 2)
- Mode registers write access (when one of the PCMCIA registers accessible by the PC is written to).

5.1.10.2 Power Down and Wake Up Using the PwrDwn Bit

The Rdy/$\overline{\text{Bsy}}$ bit is reset to zero (busy) by the PCMCIA logic if the host writes a one to the PwrDwn bit in the CCSR, and will stay in the busy state until the IMP has completed its transition to one of the low power STOP modes or can be set back to one by software (if it is not desired by the card to enter STOP mode). Once the IMP has entered low power mode, the Rdy/$\overline{\text{Bsy}}$ bit will be set to ready. When the host writes a zero to the PwrDwn bit while the IMP is in low power mode, the Rdy/$\overline{\text{Bsy}}$ bit will again be reset to busy while the IMP wakes up. IMP software should set the Rdy/$\overline{\text{Bsy}}$ bit to one when it has finished recovering from low power mode. Refer to Figure 5-7 for a more detailed description of the transition to, and wake up from low power stop mode using the PwrDwn bit.

5.1.10.3 PCMCIA Host Interrupts

The host can be interrupted by the PCMCIA controller in I/O mode through the $\overline{\text{IREQ}}$ pin which is the Rdy/$\overline{\text{Bsy}}$ pin in memory-only mode. The $\overline{\text{IREQ}}$ pin’s status is always reflected by the INTR bit in the CCSR for PCMCIA interrupts:

1. The PCMCIA controller will set the INTR bit and the $\overline{\text{IREQ}}$ pin in I/O mode if the PTIE or BERR bit in the PCHER register is set AND the PTIEn and BERRIE bits in the PCMR register are set respectively.
2. The 68000 core can set the INTR bit in the CCSR by setting the SW bit in the PCHER which will cause the $\overline{\text{IREQ}}$ pin to assert.
3. When an unmasked 16550 interrupt is issued from the 16550 controller, the INTR bit in the CCSR will be set and the $\overline{\text{IREQ}}$ pin will be asserted.

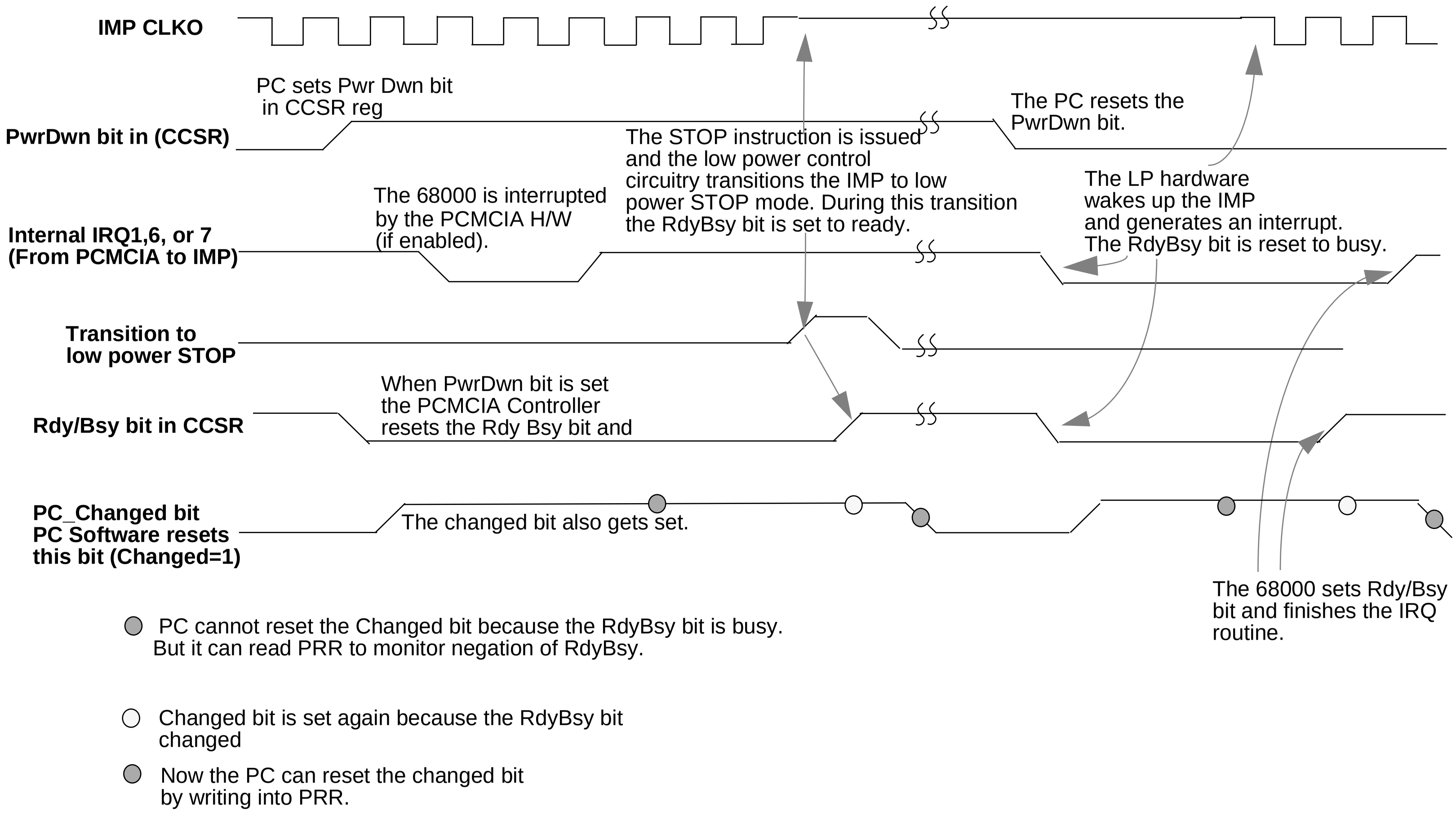

IMP CLKO
PC sets Pwr Dwn bit in CCSR reg
PwrDwn bit in (CCSR)
The PC resets the PwrDwn bit.
The STOP instruction is issued and the low power control circuitry transitions the IMP to low power STOP mode. During this transition the RdyBsy bit is set to ready.
The 68000 is interrupted by the PCMCIA H/W (if enabled).
Internal IRQ1,6, or 7 (From PCMCIA to IMP)
The LP hardware wakes up the IMP and generates an interrupt. The RdyBsy bit is reset to busy.
Transition to low power STOP
When PwrDwn bit is set the PCMCIA Controller resets the Rdy Bsy bit and
Rdy/Bsy bit in CCSR
PC_Changed bit PC Software resets this bit (Changed=1)
The changed bit also gets set.
The 68000 sets Rdy/Bsy bit and finishes the IRQ routine.
PC cannot reset the Changed bit because the RdyBsy bit is busy. But it can read PRR to monitor negation of RdyBsy.
Changed bit is set again because the RdyBsy bit changed
Now the PC can reset the changed bit by writing into PRR.

Figure 5-7. Using the PwrDwn bit to Enter and Exit Low Power Mode

5.1.10.4 The Ready Busy Signal (Rdy/Bsy)

The ready busy signal can be passed to the host through the Rdy/Bsy (IREQ in I/O mode) pin in memory only mode, and the Rdy/Bsy bit in the pin replacement register (PRR) in I/O mode.The RDY/BSY pin can be controlled by the 68000 by writing to the Rdy/Bsy bit in the PRR. The state of the Rdy/Bsy pin can be monitored in the PRR (even though the PCMCIA interface is configured as a memory only interface).

The RdyBsy bit is also controlled during the transitions to and from the three low power STOP modes. See 5.1.10.2 Power Down and Wake Up Using the PwrDwn Bit for more details.

5.2 PCMCIA PINS

5.2.1 PCMCIA Pins Supported by the MC68PM302

The pin descriptions for the MC68PM302 PCMCIA pins are given in Section 6 Signal Description**.** The MC68PM302 will read the PC_EN pin at IMP reset and if this pin is strapped to Vcc it will configure the pins with PCMCIA functionality as PCMCIA pins. If it is strapped to ground, the pins will revert to their non-PCMCIA functions.

Most PCMCIA pins have internal pullup or pulldown resistors (see 5.2.3 Pullup Control Register (PUCR)).

NOTE

The PCMCIA pins are denoted in the signal description (Section 2) as PC_*SIGNALNAME.*

5.2.2 PCMCIA Pins Not Supported

The following pins are not implemented on the MC68PM302 and should be supported with external circuitry.

Write Protect (WP)

This output signal reflects the status of the card’s write protect switch. When a card or socket is configured for the I/O interface, the write protect status may be available in the pin replacement register, and the signal is replaced in the interface by the I/O Port IS 16 bits (IOIS16) signal.

I/O IS 16 Bit Port (IOIS16)

This output signal is asserted when the address at the socket corresponds to an I/O address to which the card responds, and the I/O port being accessed is capable of 16-bit access.

When this signal is not asserted during a 16-bit I/O access, the system will generate 8-bit references to the even and odd bytes of the 16-bit port being accessed.

NOTE

The MC68PM302 PCMCIA controller does not support 16-bit I/O accesses. Therefore this pin should be tied high (not asserted) on the card.

Input Acknowledge ($\overline{\text{INPACK}}$) [I/O Operation]

This output signal is asserted when the card is selected and the card can respond to an I/O read cycle at the address on the address bus.

NOTE

If enabled for I/O mode accesses, the MC68PM302 PCMCIA controller responds to I/O read cycles at all addresses. All I/O accesses are routed to the 16550 registers. This signal may thus be asserted whenever the card enable ($\overline{\text{CE1}}$ and $\overline{\text{CE2}}$) and the $\overline{\text{REG}}$ inputs are true.

Audio Digital Waveform ($\overline{\text{SPKR}}$) [replace BVD2]

Binary audio output signal.

Card Reset (RESET)

This signal clears the card configuration option register thus placing the card in an unconfigured (memory only interface) state. A card remains in the unconfigured state until the card configuration register has been written.

In most cases this signal should trigger a reset sequence in the IMP. External circuitry such as a one shot circuit will typically be required to assure that the IMP reset is asserted long enough to meet the 68000 minimum reset specifications (see Section 7 Electrical Characteristics) when the card is reset.The card reset signal should be an input to the external reset circuitry in order to guarantee that the reset polarity and pulse times meet the requirements of the IMP.

5.2.3 Pullup Control Register (PUCR)

The PUCR contains control for the optional pullup resistors on pins with PCMCIA functionality. This register can be read and written by the 68000 core.

A set bit disables the pullup for a particular block of pins. A cleared bit enables the pullups in a particular block. Some pins have pull-ups available for non-PCMCIA functionality and some do not as noted in the bit descriptions.

PUCR 68000 Address: BAR ++$8E0

7	6	5	4	3	2	1	0
3.3VM	PUC6	PUC5	PUC4	PUC3	PUC2	PUC1	Reserved
RESET: 0	0	0	0	0	0	0	0

Read/Write

3.3VM—3.3V Mode

This bit should be set when the operating the MC68PM302 at 3.3V to ensure proper pullup resistance values.

0 = 5 V. Mode.
1 = 3 V. Mode.

PUC6

This bit, when cleared, will enable a 100K ohm pulldown resistor for each of these PCMCIA address pins. Setting this bit will disable the pulldowns.

0 = PC_A0,1,2,6,7,8,9 are pulled down
PC_A3,4,5,10,11,25 are pulled down only if PC_EN=1.
1 = All pulldowns are disabled.

PUC5

This bit, when cleared, will enable a 10K ohm pullup resistor for each of the PCMCIA control pins. Setting this bit will disable the pullups.

0 = $\overline{\text{PC\_CE1}}$, $\overline{\text{PC\_CE2}}$, $\overline{\text{PC\_REG}}$, $\overline{\text{PC\_IORD}}$ are pulled up.
1 = All pullups are disabled.

PUC4

This bit, when cleared, will enable a 10K ohm pullup resistor for these pins when PCMCIA is disabled. Setting this bit will disable the pullups.

0 = $\overline{\text{PC\_RDY}}$/R$\overline{\text{W}}$, $\overline{\text{PC\_STSCHG}}$/$\overline{\text{BR}}$, $\overline{\text{PC\_WAIT}}$/$\overline{\text{BG}}$, are pulled up only if PC_EN=0.
1 = All pullups are disabled.

PUC3

This bit, when cleared, will enable a 10K ohm pullup resistor for each of the PCMCIA control pins. Setting this bit will disable the pullups.

0 = $\overline{\text{PC\_WE}}$, $\overline{\text{PC\_IOWR}}$, $\overline{\text{PC\_OE}}$ are pulled up.
1 = All pullups are disabled.

PUC2

This bit, when cleared, will enable a 100K ohm pulldown resistor for each of the PCMCIA data pins. Setting this bit will disable the pulldowns.

0 = PC_D7-0 are pulled down only if PC_EN=1.
1 = All pulldowns are disabled.

PUC1

This bit, when cleared, will enable a 100K ohm pulldown resistor for each of the PCMCIA data pins. Setting this bit will disable the pulldowns.

0 = PC_D15-8 are pulled down only if PC_EN=1.
1 = All pulldowns are disabled.

PUC0 – Reserved.

5.3 PROGRAMMER’S MODEL

The PCMCIA controller contains a number of registers, described in the following paragraphs. Those registers can be accessed by the host and the 68000 core.

5.3.1 PCMCIA Controller Accesses

5.3.2 PCMCIA Mode Register(PCMR)

PCMR

PCMCIA Address:A25=1;$20
68000 Address: BAR + $8C0

31	30	29	28	27	26	25	24
TIEn	BERRIE	BCLROE	CLRIE	SWAP	FAST	ArbIDL	
RESET							
0	0	0	0	0	0	0	0

PCMCIA Address:A25=1;$22
68000 Address: BAR + $8C1

23	22	21	20	19	18	17	16
TValue					TPres		Ten
RESET							
0	0	0	0	0	0	0	0

PCMCIA Address:A25=1;$24
68000 Address: BAR + $8C2

15	14	13	12	11	10	9	8
RISDIS	AINC2	AINC1	Res	INTRL		16bit	ABUF
RESET							
0	0	0	0	0	0	0	0

PCMCIA Not Accessible
68000 Address: BAR + $8C3

7	6	5	4	3	2	1	0
Res	Res	PC_DREQEN	PCDSWP	PCDDIR		IDOI	PCSTR
RESET							
0	0	0	0	0	0	0	0

Read/Write

ArbIDL—Arbitration IDLE Time

These bits are used to indicate when to release the 68000 bus after a common memory space transfer has terminated.

NOTE

> This mechanism is implemented only for PCMCIA common memory space accesses that are transferred into the 68000 bus. Attribute space cycles are always made on a cycle steal basis. I/O space cycles are mode to the 16550 registers only.

00 = Cycle steal mode. The 68000 bus is released after the cycle has been terminated. The arbitration logic will arbitrate for the 68000 bus when next cycle starts (CE active)

01 = Burst mode. The arbitration logic waits for 4 idle 68000 bus clocks on the PCMCIA bus before releasing the 68000 bus.

10 = Burst mode. The arbitration logic waits for 8 idle 68000 bus clocks on the PCMCIA bus before releasing the 68000 bus.

11 = Burst mode. The arbitration logic waits for 16 idle 68000 bus clocks on the PCMCIA bus before releasing the 68000 bus.

FAST—Fast Burst Mode

Setting this bit enables fast, burst mode PCMCIA common memory accesses without the WAIT signal (except for the first access). When the PCMCIA controller is programmed to burst mode by setting the ArbIDL1-0 bits to non-zero, the burst can be done in two different modes. The first cycle with arbitration is identical in both modes.

0 = Normal mode

When the PCMCIA controller is granted control of the 68000 bus and the PC initiates a common memory space cycle, the PCMCIA controller asserts the $\overline{\text{WAIT}}$ signal on the PCMCIA line. It then generates a 68000 cycle. When the cycle is terminated with $\overline{\text{DTACK}}$, data is transferred to the PCMCIA data lines and the $\overline{\text{WAIT}}$ signal is negated.

1 = Fast mode

This mode is used to minimize the usage of the $\overline{\text{WAIT}}$ signal on the PCMCIA interface, allowing fast direct access transfers to be performed. In this mode, when the PCMCIA controller is granted control of the 68000 bus and the PC initiates a common memory space cycle, the PCMCIA controller will not assert the $\overline{\text{WAIT}}$ signal on the PCMCIA line. The cycle on the 68000 bus will use the memory interface signals $\overline{\text{WE}}$ and $\overline{\text{OE}}$, but not the $\overline{\text{DS}}$. The cycle will be terminated when the PCMCIA cycle is terminated. The PCMCIA controller timing ignores $\overline{\text{DTACK}}$ and $\overline{\text{BERR}}$ so if 68000 bus cycles are not complete when the PCMCIA bus cycle is terminated garbage data will result (68000 Bus Errors can be reported to the host by setting the BERRIE bit in the PCMR). Fast mode accesses are only allowed for external memory space accesses. Accesses to internal memory space will result in erratic operation.

INTRL—Interrupt Level

The PCMCIA control logic will generate an interrupt to the 68000 core when the host writes one of the configuration registers if its interrupt mask bit is set in the PCMCIA Configuration Registers Write Mask Register (PCRWMR). The interrupt level generated as a result of one of the accesses can be programmed to the following 68000 interrupt levels:

00 = No interrupt is generated.
01 = The interrupt is generated on level 1.
10 = The interrupt is generated on level 6.
11 = The interrupt is generated on level 7.

NOTE

The interrupt will be generated as level triggered. The user should program the GIMR register to generate the interrupt vector.

BCLROE—Bus Clear Output Enable

When this bit is set the PCMCIA controller will assert the BCLR signal when requesting the bus and it is not granted. The IDMA, when programed to work under software control, will release the bus at the end of its current transfer.

BCLRIE—Bus Clear Input Enable

When this bit is set, the PCMCIA controller will release the 68000 bus when BCLR is asserted and the current cycle is terminated, regardless of the ArbIDL value. BCLR can be asserted for the following reasons:

1. The SDMA requests the 68000 bus.
2. An external master requests the 68000 bus.
3. A 68000 interrupt is pending and the BCLM bit is set in the SCR.

BERRIE—Bus Error Interrupt Enable

When this bit is set, the PCMCIA control logic will generate an interrupt to the PC when a PCMCIA bus cycle results in a 68000 bus error.

SWAP—Swapper Enable

This bit, when set, enables the swapping function for direct read and write PCMCIA accesses. The PCMCIA logic swaps the high and low bytes of 16 bit words when doing direct read or write accesses. Both the PCMCIA interface and the 68000 bus must be set to 16-bit wide for the swapper to be enabled.

TEn—Timer Enable

The PCMCIA direct access mechanism has a bus monitor (hardware watchdog) timer to prevent excessively long cycles, or hung state conditions on the PCMCIA interface. The timer is activated when a PCMCIA interface cycle begins. The WAIT signal is negated and a maskable interrupt is generated when the timer expires before the cycle is terminated.

0 = The PCMCIA interface bus monitor timer is disabled.
1 = The PCMCIA interface bus monitor timer is enabled.

TIEn—PCMCIA Watchdog Timer Interrupt Enable

This bit, when set, enables interrupt generation to the PC from the PCMCIA bus monitor timer. A PTIE interrupt is generated to the PC when this bit is set and the PCMCIA bus monitor timer expires.

TPres—PCMCIA Watchdog Timer Prescaler

The PCMCIA watchdog timer uses this prescaler value. The PCMCIA watchdog timer uses the prescaler output clock as an input to its 5-bit down counter.

00 = Divide by 1.
01 = Divide by 4.
10 = Divide by 8.
11 = Divide by 16.

TValue—Timer Value

The PCMCIA watchdog timer is a 5-bit down counter with the programmable TValue preset. The counter has a programmable prescaler. The counter will count the $\overline{\text{WAIT}}$ length and, if expired, the PCMCIA cycle will be terminated and the PTIE interrupt to the PC will be generated (if enabled).

ABUF—Address Buffer Control

This bit, when set, enables the address buffer control output function. The PCMCIA controller will not drive the high address lines (A12-A23) as programmed in the base registers but instead will drive an address buffer control line with the appropriate timing. This mechanism is useful when the paging is not sufficient for the system design requirements. External address buffers must be implemented on the card.

16Bit—PCMCIA Interface is 16-Bit

0 = The PCMCIA interface is 8 data bits wide. The D8-D15 and CE2 pins can be muxed for their alternative functions.
1 = The PCMCIA interface is 16 data bits wide.

AINC1,2—Auto Increment

When set, the corresponding common memory base address register will auto-increment when the last entry in the page is accessed. The last entry differs depending on the PCMCIA bus width programmed in bit 16 in the PCMCR register. For PCMCIA 8 bit mode (16bit=0 in PCMR) the last entry is PC_A11:PC_A0=fff; for PCMCIA 16 bit mode (16bit=1 in PCMR) the last entry in the page is PC_A11:PC_A1=7ff, and CE2_=0 (word access) or PC_A0=1 (byte access). This mechanism is useful during burst mode accesses because it eliminates the need to reprogram the base address register when crossing a page boundary.

RISWDIS—Ring Indicate Software Disable

0 = The RIEvt bit in the IOEIR register will be set when $\overline{\text{RI}}$ is asserted.
1 = The RIEvt bit in the IOEIR is not set when $\overline{\text{RI}}$ is asserted. This mechanism is useful to allow the 68000 core to either integrate or debounce the $\overline{\text{RI}}$ signal by software before updating the RIEvt bit (the $\overline{\text{RI}}$(PB9) pin can be read by the 68000 core

through the port B data register). Refer to Figure 5-6. Ring Indicate (PB9) Connection Options.

For bits PCSTR, IDOI, PCDDIR, PCDSWP, PC_DREQEN (See 5.3.11 PCMCIA DMA Support)

5.3.3 PCMCIA Configuration Registers Write Event Register (PCRWER)

PCRWER

PCMCIA Address: Not accessible
68000 Address: BAR + $8C4

7	6	5	4	3	2	1	0
RES3	RES2	ES1	IOIEIR	SCR	PRR	CCSR	COR
RESET							
0	O	0	0	0	0	0	0

Read/Write

The PCMCIA configuration registers write event register is an 8-bit register that reports PC writes to one of the PCMCIA configuration registers. The appropriate event bit is set depending on the location of host write, regardless of the corresponding mask bit in the PCRWMR. If the corresponding mask bit is set and the INTRL0-1 bits in the PCMR register are non-zero, an interrupt will be generated. In order to generate an interrupt, the INTRL0-1 bits in the PCMR register must be set to a non-zero value. The PCRWER is a memory mapped register that may be read at any time. A bit can be reset by writing a one and is left unchanged by writing a zero. More than one bit may be reset at a time. All bits are cleared by reset.

CCR—Card Configuration Register

This bit is set when the PC writes to the card configuration option register. The 68000 core may monitor writes to this register to detect SRESET and changes to the configuration index field (to determine if the host has enabled or disable the I/O card). This bit can be cleared by writing a one to it.

CCSR—Card Configuration and Status Register

This bit is set when the PC writes to the card configuration and status register. The 68000 core may monitor writes to this register to detect if the PC has requested to place the card in the power-down state by setting the PwrDwn bit. This bit can be cleared by writing a one to it.

PRR—Pin Replacement Register

This bit is set when the PC writes to the pin replacement register. This bit can be cleared by writing a one to it.

SCR—Socket and Copy Register

This bit is set when the PC writes to the socket and copy register. This bit can be cleared by writing a one to it.

IOEIR—IO Event Indication Register

This bit is set when the PC writes to the IO event indication register. This bit can be cleared by writing a one to it.

Res1—Reserved1 Register

This bit is set when the PC writes to the reserved1 register. This bit can be cleared by writing a one to it.

Res2—Reserved2 Register

This bit is set when the PC writes to the reserved2 register. This bit can be cleared by writing a one to it.

Res3—Reserved3 Register

This bit is set when the PC writes to the reserved3 register. This bit can be cleared by writing a one to it.

5.3.4 PCMCIA Configuration Registers Write Mask Register (PCRWMR)

The PCRWMR is 8-bit memory-mapped, read-write register that has the same bit format as the PCRWER. If a bit in the PCRWMR is a one, the corresponding interrupt in the PCRWER will be enabled. If the bit is zero, the corresponding interrupt in the PCRWMR will be disabled. PCRWMR is cleared at reset.

5.3.5 PCMCIA Access Wake-Up Event Register(PCAWER)

PCAWER

PCMCIA Address: Not accessible
68000 Address: BAR + $8C5

7	6	5	4	3	2	1	0
PCDRDY	PCTC	–	MRW	CMS2	CMS1	ATS	IOS
RESET							
0	O	0	0	0	0	0	0

Read/Write

The 68000 core may program the IMP into stand-by mode when there is no activity on the card for long periods of time. PC accesses to the card will wake-up the IMP.

The PCAWER is an 8-bit register used to report which type of PC write access was made to the card to wake the IMP up from STAND-BY mode. The 68000 core will receive an interrupt if:

1. It is awakened by a PCMCIA access, and

2. The INTRL bits in the PCMR are non-zero (PCMCIA interrupts enabled), and

3. The corresponding mask bit for the access type is set in the PCAWMR.

NOTE

Read accesses by the PC to one of the PCMCIA configuration registers will **not** wake up the IMP.

The appropriate event bit will be set, regardless of the corresponding mask bit in the PCRWMR. If the corresponding mask bit is set, an interrupt will be generated. The PCAWER is a memory mapped register that may be read at any time. A bit is reset by writing a one and is left unchanged by writing a zero. More than one bit may be reset at a time, and the register is cleared by reset.

IOS—IO Space Access

This bit is set when the PC accesses the 16550 emulation module (an I/O space access is made). When the IMP is in stand-by mode, the IMP will resume its operation. An interrupt may be generated if the corresponding mask bit is set. This bit is cleared by writing one.

ATS—Attribute Space Access

This bit is set when the PC accesses the CIS memory. When the IMP is in stand-by mode, the IMP will resume its operation. An interrupt may be generated if the corresponding mask bit is set. This bit is cleared by writing one.

NOTE

PC writes to one of the PCMCIA configuration registers while the IMP is in stand-by mode will also resume operation. The 68000 will be able to detect it by the 5.3.5 PCMCIA Access Wake-Up Event Register(PCAWER).

CMSI1—Common Memory Space Access

This bit is set when the PC accessing common memory space with A25=1 (CMBAR1 is selected). When the IMP is in stand-by mode, the IMP will resume its operation. An interrupt may be generated if the corresponding mask bit is set. This bit is cleared by writing one.

CMSI2—Common Memory Space Access

This bit is set when the PC accessing common memory space with A25=0 (CMBAR2 is selected). When the IMP is in stand-by mode, the IMP will resume its operation and an interrupt may be generated if the corresponding mask bit is set. This bit is cleared by writing one.

MRW—Mode Registers Write.

This bit is set when the PC writes to one of the PCMCIA mode or base registers in the attribute space. When the IMP is in stand-by mode, the IMP will resume its operation. An interrupt may be generated if the corresponding mask bit is set. This bit is cleared by writing one.

Bits PCTC and PCDRDY are described in 5.3.11 PCMCIA DMA Support

5.3.6 PCMCIA Access Wake-up Mask Register(PCAWMR)

The PCAWMR is an 8-bit memory-mapped, read-write register that has the same bit format as the PCAWER. If a bit in the PCAWMR is a one, the corresponding interrupt in the PCAW-

ER will be enabled. If the bit is zero, the corresponding interrupt in the PCRWMR will be disabled. PCAWMR is cleared at reset.

Bits PCTC and PCDRDY are described in Section 5.3.11

5.3.7 PCMCIA Host (PC) Event Register(PCHER)

PCHER

PMCIA Address:A25 = 1; $2A
68000 Address: BAR + $8C8

7	6	5	4	3	2	1	0
–	–	–	–	–	SW	PTIE	BERR
RESET							
0	O	0	0	0	0	0	0

Read/Write

BERR—Bus Error

This bit is set when a cycle is terminated by bus error. An interrupt to the PC will be generated if the BERRIE bit is set. This bit is cleared by writing one.

PTIE—PCMCIA Timer Expired

This bit is set when a cycle is terminated by a PCMCIA watchdog timer expiration. An interrupt to the PC will be generated if the TIEn bit is set. This bit is cleared by writing a one.

SW—68000 Software Interrupt

The 68000 S/W may set this bit to generate an interrupt to the PC. This bit is cleared by the PC writing a one.

5.3.8 CIS Base Address Register(CISBAR)

CISBAR

PCMCIA Address: A25 = 1; $30
68000 Address: BAR + $8CC

15	14	13	12	11	10	9	8
BA23	BA22	BA21	BA20	BA19	BA18	BA17	BA16
RESET				0			
0	0	0	0		0	0	0

PMCIA Address: A25 = 1; $32

7	6	5	4	3	2	1	0
BA15	BA14	BA13	BA12	FC2	FC1	FC0	V
RESET							
0	0	0	0	0	0	0	0

Read/Write

This register is used to locate PCMCIA CIS accesses to the 68000 bus. The CIS must be located in memories (RAM, EEPROM etc.) on the 68000 bus. The 68000 core should program the starting address of the CIS memory structure into this register.

NOTE

Since the PCMCIA attribute space accesses are only 8-bits and A0 should be always 0, the CIS memory should be on the 68000 even addresses only.

NOTE

This register is in the PCMCIA attribute space and this 24-bit register address will be 3 bytes on **even** addresses only.

V—Valid Bit

This bit indicates that the contents of the base register is valid. The PCMCIA attribute space cycle ($\overline{\text{REG}}$ is asserted and A25=0) will not be transferred to the 68000 bus until the V-bit is set.

0 = This CIS base address register is invalid.
1 = This CIS base address register is valid.

BA12-23—Base address

The base address field should contain the upper address bits of the CIS data structure address on the 68000 bus. The lower address bits will be taken from the PCMCIA address lines.

FC0-2—Function Code

The function code field should contain the function code of the CIS data structure address on the 68000 bus.

5.3.9 Common Memory Space Base Address Register (CMBAR1,2)

CMBAR1, CIMBAR2

PCMCIA Address: A25 = 1; $38, $40
68000 Address: BAR + $8D0, $8D4

15	14	13	12	11	10	9	8
BA23	BA22	BA21	BA20	BA19	BA18	BA17	BA16
RESET							
0	0	0	0	0	0	0	0

PCMCIA Address: A25 = 1; $3A, $42

7	6	5	4	3	2	1	0
BA15	BA14	BA13	BA12	FC2	FC1	FC0	V
RESET							
0	0	0	0	0	0	0	0

Read/Write

There are two identical registers to map common memory accesses into the 68000 bus. This mechanism will enable the card designer to use memories (SRAM, DRAM etc.) or peripherals on the 68000 bus and access them by the host. The 68000 core or the PC should program the starting 68000 address of the memory or peripheral into these registers. A25 will distinguish between the two base registers. When A25=1 CMBAR1 will be selected and when A25=0 CMBAR2.

NOTE

Since the PCMCIA attribute space accesses are only 8-bits, this 16-bit register address will be 2 bytes on even addresses only.

V—Valid Bit

This bit indicates that the contents of the base register is valid. The PCMCIA common memory space cycle will not be transferred into the 68000 bus until the V-bit is set.

0 = This register base address is invalid.
1 = This register base address is valid.

BA12-23—Base Address

The base address field should contain the upper address bits of the I/O memory address on the 68000 bus. The lower address bits will be taken from the PCMCIA address lines.

NOTE

The 2Kbyte block defined by this base address should not contain any internal addresses (internal dual-port RAM, internal registers, etc.) if it is desired to use the fast mode accesses. Fast mode accesses (Fast Mode=1) to internal memory locations will result in erratic operation.

FC0-2—Function Code

The function code field should contain the function code of the common memory address on the 68000 bus.

5.3.10 Card Configuration Registers

5.3.10.1 Configuration Option Register - COR

COR

PCMCIA Address: A25 = 1; $0
68000 Address: BAR + $870

7	6	5	4	3	2	1	0
SRESET	LevIREQ	Configuration Index					
0	0	0	0	0	0	0	0

Read/Write

The 68000 core will receive an interrupt when this register is written by the PC. See 5.3.5 PCMCIA Access Wake-Up Event Register(PCAWER).

SRESET—Reset Card

When this bit is set to one, the 68PM302 enters a hardware reset state. The $\overline{\text{RESET}}$ pin is asserted and the CPU is reset. The 68PM302 stays in the reset state until the PCMCIA resets the bit.

This bit is also cleared by asserting $\overline{\text{RESET}}$ and $\overline{\text{HALT}}$ simultaneously.

LevIREQ—Level Mode Interrupts

The MC68356 PCMCIA controller supports the interrupt level mode only. (Regardless of this bit value)

NOTE

The pulse mode may be supported by the 68000 software.

Configuration Index

This field is written with the index number of the entry in the card's configuration table which corresponds to the configuration which the system chooses for the card. When the configuration index is 0, the card's I/O is disabled and will not respond to any I/O cycles, and will use the memory only interface.

5.3.10.2 Card Configuration and Status Register (CCSR)

CCSR

PCMCIA Address: A25 = 1; $2
68000 Address: BAR + $871

7	6	5	4	3	2	1	0
Changed	SigChg	IOis8	RingEn	Audio	PwrDwn	Intr	Res(0)

Read/Write

The 68000 core will receive an interrupt when this register is written by the PC. See 5.3.5 PCMCIA Access Wake-Up Event Register(PCAWER).

Changed

This bit indicates that one or more of the pin replacement register bits CBVD1, CBVD2, CRdy$\overline{\text{BSY}}$, CWProt is set to one or when one or more of the I/O event indication register event bits are set and its corresponding enable bit is also set. When this bit is set, the $\overline{\text{STSCHG}}$(BVD1) pin is held low if the SigChg bit is 1 and the card is configured as I/O.

SigChg

1 = When the card is configured for I/O interface, the changed bit controls the $\overline{\text{STSCHG}}$ signal and is called the changed status signal.
0 = The BVD1 ($\overline{\text{STSCHG}}$) line will be held high while the card is configured as I/O.

RingEn

0 = The state of the $\overline{\text{STSCHG}}$ is controlled by the SigChg bit as described above.
1 = The $\overline{\text{STSCHG}}$ signal is used to indicate the state of the ring indicate signal. $\overline{\text{STSCHG}}$ is asserted (low) whenever ringing is present on the RI (PB9) pin. $\overline{\text{STSCHG}}$ is negated (high) when there is no ringing.

IOis8

1 = The host can provide I/O cycles only with 8-bit D0-D7 data path. Accesses to 16-bit registers will occur as two byte accesses.
0 = The host can provide I/O cycles with 16-bit data path.

Audio

This bit is set to one to enable audio information on BVD2 pin while the card is configured.

PwrDwn

This bit is set to one to request the card enter a powerdown state. The 68000 core will receive an interrupt (if enabled) when this register is written (see 5.3.4 PCMCIA Configuration Registers Write Mask Register (PCRWMR)) and should check the state of this bit to determine if the PC has placed the card into power down mode. If it has, the 68000 should place the card into power down mode. When this bit is reset while the IMP is in any of the three STOP modes, the 68000 core will be waked up (the PC can access the configuration registers while in powerdown mode).

Intr

This bit represents the state of the interrupt request. This value is available whether or not interrupts have been configured. This signal should remain true until the condition which caused the interrupt request has been serviced. See 5.1.10.3 PCMCIA Host Interrupts for more information on the functionality of this bit.

5.3.10.3 Pin Replacement Register Organization (PRR)

PRR

PCMCIA Address: A25 = 1; $4
68000 Address: BAR +$872

7	6	5	4	3	2	1	0
CBVD1	CBVD2	CRdy/Bsy	CWProt	RBVD1	RBVD2	RRdy/Bsy	RWProt

Read/Write

The 68000 core will receive an interrupt when this register is written by the PC. See 5.3.5 PCMCIA Access Wake-Up Event Register(PCAWER).

CBVD1,CBVD2

These bits are set to one when the corresponding bits change state. These bits may also be written by the host.

CRdy/Bsy

This bit is set to one when the corresponding bit changes state. These bits may also be written by the host.

CWProt

This bit is set to one when the corresponding bit changes state. These bits may also be written by the host.

RBVD1, RBVD2

When read these bits represent the internal state of the corresponding signals. When this bit is written as one, the corresponding changed bit is also written.Since the MC68PM302 does not support this pin directly, external circuitry should be used to implement this pin.

NOTE

The 68000 should write the state of this signal into this location. The corresponding changed bit will be updated according to the written data.

RRdy/Bsy

When read, this bit represents the internal state of the corresponding signal. When this bit is written as one, the corresponding changed bit is also written. See 5.1.10.4 The Ready Busy Signal (Rdy/Bsy) for more information on the functionality of this bit.

NOTE

The 68000 core should write the state of this signal into this location. The corresponding changed bit will be updated according to the written data.The 68000 core should set this bit to one after initializing the PCMCIA controller to signal ready (high) to the host. When set to zero, the PCMCIA controller will signal busy. After reset, the PCMCIA controller will set the RRdy/$\overline{\text{Bsy}}$ pin low and clear this bit.

NOTE

In I/O mode the RDY/$\overline{\text{BSY}}$ pin becomes the $\overline{\text{IREQ}}$ pin.

RWProt

When read, this bit represents the internal state of the corresponding external signal. When this bit is written as one, the corresponding changed bit is also written. Since the MC68PM302 does not support this pin directly, external circuitry should be used to monitor the state of this signal.

NOTE

The 68000 should write the state of this signal into this location. The corresponding changed bit will be updated according to the written data.

5.3.10.4 Socket and Copy Register (SCR)

SCR

PCMCIA Address: A25 = 1; $6
68000 Address: BAR + 873

7	6	5	4	3	2	1	0
Res	Copy Number			Socket Number			

Read/Write

This register can be R/$\overline{\text{W}}$ accessed by the PC and the 68000 core. The 68000 core will receive an interrupt when this register is written by the PC. See 5.3.5 PCMCIA Access Wake-Up Event Register(PCAWER).

NOTE

This is an optional register which the card may use to distinguish between similar cards installed in a system.

5.3.10.5 I/O Event Indication Register (IOEIR)

IOEIR

PCMCIA Address: A25 = 1; $8
68000 Address: BAR + $874

7	6	5	4	3	2	1	0
ResEvt3	ResEvt2	Pievt	RIEvt	ResEnab3	ResEnab2	PIEnab	RIEnab

Read/Write

The 68000 core will receive an interrupt when this register is written by the PC. See 5.3.5 PCMCIA Access Wake-Up Event Register(PCAWER).

This is an optional register that contains information about the changes in the card status. Its bit assignments are defined below. This register may be read or written. The upper four bits are latched to a one when the corresponding IO event occurs on the PC card. When one of these upper four bits is latched and the corresponding enable bit in the lower nibble has also previously been set, the changed bit in the card configuration and status register (CCSR) is set to a one, and the $\overline{\text{STSCHG}}$ pin will be driven low (if enabled by the SigChg bit in the CCSR).

The host can clear any one of the upper four bits by writing a one to that bit. Writing a zero to these bits will have no effect. All bits of this register are cleared by an IMP reset.

ResEvt3

Reserved for future expansion/definition. Set to zero.

ResEvt3

Reserved for future expansion/definition. Set to zero.

PIEvt

This bit is latched to a one by the card (the 68000 core can set this bit) after the receipt of a valid incoming packet over a modem channel. When this bit is set to a one and the PI-Enab bit is set to a one, the changed bit in the card configuration and status register will also be set to a one. If the SigChg bit in the card configuration and status register has also been set by the host, then the $\overline{\text{STSCHG}}$ pin will be driven low. Writing one to this bit will clear it to zero. Writing a zero to this bit has no effect.

RIEvt

This bit is latched to a one by the MC68PM302 on the start of each cycle of the ring frequency on the phone line. When this bit is set to a one, and the RIEnab bit will be set to a one the changed bit in the card configuration and status register will also be set to a one. If the SigChg bit in the card configuration and status register (CCSR) has also been set by the PC, then the $\overline{\text{STSCHG}}$ pin will be driven low. Writing one to this bit will clear it to zero. Writing a zero to this bit has no effect.

NOTE

If it is desired to have the 68000 core debounce or otherwise process the RI signal before passing it to the host, the RISDIS bit can be set in the PCMR register which will disconnect the direct connection to the RIEvt bit (See Figure 5-6).

ResEnab3

Reserved for future expansion/definition. Set to zero.

ResEnab2

Reserved for future expansion/definition. Set to zero.

PIEnab

Setting this bit to a one enables setting the changed bit in the card configuration and status register when the PIEvt bit is set. When this bit is cleared to a zero, this feature is disabled. The state of PIEvt bit is not affected by this bit.

RIEnab

Setting this bit to a one enables setting the changed bit in the card configuration and status register when the RIEvt bit is set. When this bit is cleared to a zero, this feature is disabled. The state of RIEvt bit is not affected by this bit.

5.3.10.6 Reserved Registers

The MC68PM302 implements additional 3 card configuration registers. Those 8-bits registers can be read/written by both the 68000 core and the PC. They are reserved for future use.

5.3.11 PCMCIA DMA Support

The PCMCIA Controller supports DMA transfers between the PC Host and PCMCIA Card. Compliant with DMA Change Pages PCMCIA document 0035 Release 004.

5.3.11.1 PC DMA Operation Description

As a preface, it useful to clear right away that in the PCMCIA specification on a “DMA Read cycle” the data flows from the PC Host to the PC Card (PC_IOWR_ is asserted); on a “DMA Write cycle” the data flows from the PC Card to the PC Host (PC_IORD_ is asserted).

On the DMA transfers the data between the PC Host and the Card is latched internally in the chip in the 16 bit PCDMAD (PC DMA Data) Register. The Data transfer to the 68k bus can be performed by the 68k CPU (Interrupt driven) or by the on chip IDMA (according to IDOI bit in PCMR register).

NOTE

The PC DMA Logic is not affected by the contents of the COR register.

5.3.11.1.1 Interrupt/Polling Mode:

In this mode the 68k CPU can either poll PCDRDY bit, or enable its interrupt. When PCDRDY event occurs, the 68K CPU transfers data to/from PCDMAD Register. The PCDRDY event is reset by writing 1 to the bit.

NOTE

Accessing the PCDMAD Register by the 68k when PCDRDY=0 may result in erroneous behavior.

5.3.11.1.2 IDMA Mode:

Software programs the IDMA in the chip to external request (edge sensitive). For DMA Read cycles, SAPI=0 and the Source Operand Address is the address of PCDMAD (PC DMA Data mapped at Base+878) Register; for DMA Write cycles DAPI=0 and the Destination Operand Address is the address of PCDMAD Register. The byte count has to be equal or longer to the transfer length.

Software enables PC_DMA and data flow direction in PCDMA Register. When the PCSTR bit is set by software the PC_DMA hardware starts transfers. If the data flows from the PC Host to the 68k bus (DMA Read cycle), the chip asserts PC_DREQ_ to the PC Host. The PC Host executes a DMA Read cycle to the card. The data is latched in the chip. The chip then internally asserts DREQ_ signal to the IDMA. The IDMA executes a source operand read from PCDMAD and a Destination operand write to Memory or External IO device on 68k bus. If the data flows from the PC Card to the PC Host (DMA Write cycle), the chip internally asserted DREQ_ signal to the IDMA. The IDMA executes a source operand read from the 68k bus (Memory or IO device) and a destination operand write to PCDMAD register. The chip then asserts PC_DREQ_ to the PC Host, which in turn it executes a PC DMA Write cycle to the card.

The block transfer is terminated in two ways: If the PC turns on during a transfer, PC_DREQ_ or IDMA DREQ_ are not asserted any more, and an interrupt is issued to the 68k if enabled (event is in PCAWER Register). If the IDMA Transfer Count is exhausted it internally asserts DONE_ and no more PC_DREQ_ or IDMA DREQ_ will be issued, until the IDMA and the PC DMA registers are reprogrammed. PCSTR is automatically reset upon termination. In addition to those hardware events, the CPU can reset the PCSTR bit and terminate the transfer.

NOTE

After software reset of the PCSTR, the PC DMA hardware completes the transfer.

5.3.11.2 Data Path and Swapping Issues:

The PC DMA specification does not define a byte transfer on PCD15:8. Byte transfers occur only on PCD7:0. With this constrain we assume that the specification is tailored only for Intel type masters. Furthermore, we assume that a PC Host with 16 bit data bus, does not initiate a DMA block transfer at an odd address (by doing so the Host would have to implement byte swapping from its high byte to its low byte).

The following combinations of data bus width are supported:

1. 68k-8bit PC-8bit
2. 68k-16bit PC-16bit
3. 68k-16bit PC-8bit

In all systems, the user programs the DMA Operand Address for the PC to 878 hex.

1. In this system, BUSW pin = 0, and 16BIT = 0 in the PCMR register. The PCDSWP bit in PCMR is not used (don't care). The data flows from ID7:0 to PCD7:0.

2. In this system, BUSW pin = 1 and 16BIT = 1 in the PCMR register. If the data is word oriented, there is no need to swap it; therefore PCDSWP = 0 in PCMR (Note that in this case no byte transfers occur). If the data transferred is byte oriented, the user needs to swap it (PCDSWP = 1 in PCMR). PCDMAD bits 7:0 are transferred into ID bits 15:8, and PCDMAD bits 15:8 are transferred into ID bits 7:0.

If an uneven number of bytes is transferred, The data on ID7:0 is swapped to PCD15:8. (Note that according to our assumption above, all accesses in 16 bit systems are aligned to even addresses).

3. In this system BUSW pin = 1 and 16BIT = 0 in the PCMR register. The user programs the operand size for the PC and for the 68k to 8 bits (SSIZE or DSIZE) and PCDSWP = 1 in PCMR. The data flows from ID15:8 to PCD7:0. In Interrupt/Polling mode, the CPU has to perform packing and unpacking of the data.

5.3.11.3 Description of Pin Reassignment

For transfers from the PC Host to the 68K bus, DMA support requires replacement of I/O interface pins with DMA signals. This specification describes the replacements as they apply to the MC68PM302; however, it is up to the user to connect the pins of the chip to the correct signals on the PCMCIA connector.

The pins of the MC68PM302 are redefined as follows (only redefined pins that affect present specification are mentioned here; left-most functions are new):

1. IDREQ_/PC_DREQ_/PB8 (PCMCIA DMA Request) in PC DMA mode this pin is an output; the user should connect it either to SPKR or INPACK or IOIS16 signals on the PCMCIA Connector.

5.3.11.4 Description of Registers for PC_DMA

The following sections describe the registers used for the PC_DMA.

5.3.11.4.3 PCMCIA Access Wake-Up Event Register (PCAWER)

bit 6 - PCTC (PCMCIA Terminal Count)

0 No event has occurred

1 PCTC (PCMCIA Terminal Count) This event is asserted if the PC Host asserted TC signal on the PC DMA transfer (bit is reset by writing one)

bit 7 - PCDRDY (PCDMAD Register Ready)

0 PCDMAD register is busy
1 PCDMAD register is ready for 68K bus
During "DMA Write cycle": Bit is set when data is read from PCDMAD (by PC Host)
During "DMA Read cycle": Bit is set when data is written into PCDMAD (by PC Host). In both cases, bit is reset by writing 1 to it.

NOTE

On "DMA Read cycle" (PC_WE_ active), if Terminal Count is active, both PCTC and PCDRDY bits are set.

On "DMA Write cycle" (PC_OE_ active) if Terminal Count is active, only PCTC bit is set.

5.3.11.4.4 PCMCIA Access Wake-Up Mask Register (PCAWMR)

bit 6 - PCTC (PCMCIA Terminal Count Mask)

0 Mask interrupt to CPU
1 Do not mask interrupt

bit 7 - PCDRDY (PCDMAD Register Busy Mask)

This bit is automatically reset when a block transfer is terminated.

0 Mask interrupt to CPU
1 Do not mask interrupt

5.3.11.4.5 PCMCIA Mode Register (PCMR) (Additional Bits)

bit 0 - PCSTR (PCMCIA DMA Start Operation)

0 Stop PC DMA Controller
1 Start the operation on the PC DMA controller

bit 1 - IDOI (IDMA Mode or Interrupt Mode)

0 IDMA mode
1 68k Interrupt mode

bit 2 - PCDDIR (PCMCIA DMA Direction)

0 'DMA write cycle' the data flows from the PC card to the PC host
1 'DMA read cycle' the data flows from the PC host to the PC card

bit 3 - PCDSWP (PCMCIA Data Swap)

0 Do not swap data bytes
1 Swap data bytes

bit 4 - PC_DREQEN (PCMCIA DMA Request Enable)

0 PC_DREQ_/IDREQ_/PB8 pin is connected to IDMA PBIO8 and Interrupt Controller The pin functions as interrupt if unmasked in interrupt controller, as IDREQ_ if

IDMA is programmed for external request, and as Parallel IO. Its direction is determined by PBIO register programming.

1 PC_DREQ_/IDREQ_/PB8 pin is output and functions as PC_DREQ_. The user should connect the pin either to SPKR or INPACK or IOIS16.

NOTES:

Do not program PC_DREQEN=1 and PCSTR=1, during the same write cycle. First program PC_DREQEN=1, then PCSTR=1. (Software sequence will be provided).

If pin PC_EN=0, the DREQ_ to the IDMA is connected to PA13/ DREQ_ pin and this pin acts as PB8 (parallel IO, or Interrupt if PIT is disabled).

5.3.11.4.6 PCMCIA DMA Data Register (PCDMAD) (Mapped at Address Base+$878)

The PCDMAD is a 16 bit register that buffers the data between the PC Host and the PC Card on PC DMA transfers. PCMCIA 8 bit transfer (PC_CE2=1, pc_CE1=0) writes data to D7-0 of PCDMAD Register. If PCDSWP is set the DMA operand address for PCDMAD has to be base+$878; if reset the DMA operand address for the PCDMAD has to be base+$879.

SECTION 6
SIGNAL DESCRIPTION

This section defines the MC68PM302 signals (Figure 6-1). For more detail on each signal, refer to the paragraph named for the signal.

6.1 IMP PINS

Some of the pins allotted to the IMP portion of the MC68PM302 are multiplexed between the normal IMP signals and the PCMCIA signals. The pin multiplexing is organized so the user has all the corresponding IMP pins available when not using PCMCIA. In fact, when not using the PCMCIA interface, additional PIO ports are available. The following signals are listed in two functional groups, because they are sampled at reset and change function after reset: TRIS/TIN2/PB5, BUSW/PC_ABUF, MODCLK/BRG3/PA12, PC_EN/$\overline{\text{TOUT2}}$/PB6, $\overline{\text{BOOT}}$/BRG2/SDS2/PA7.

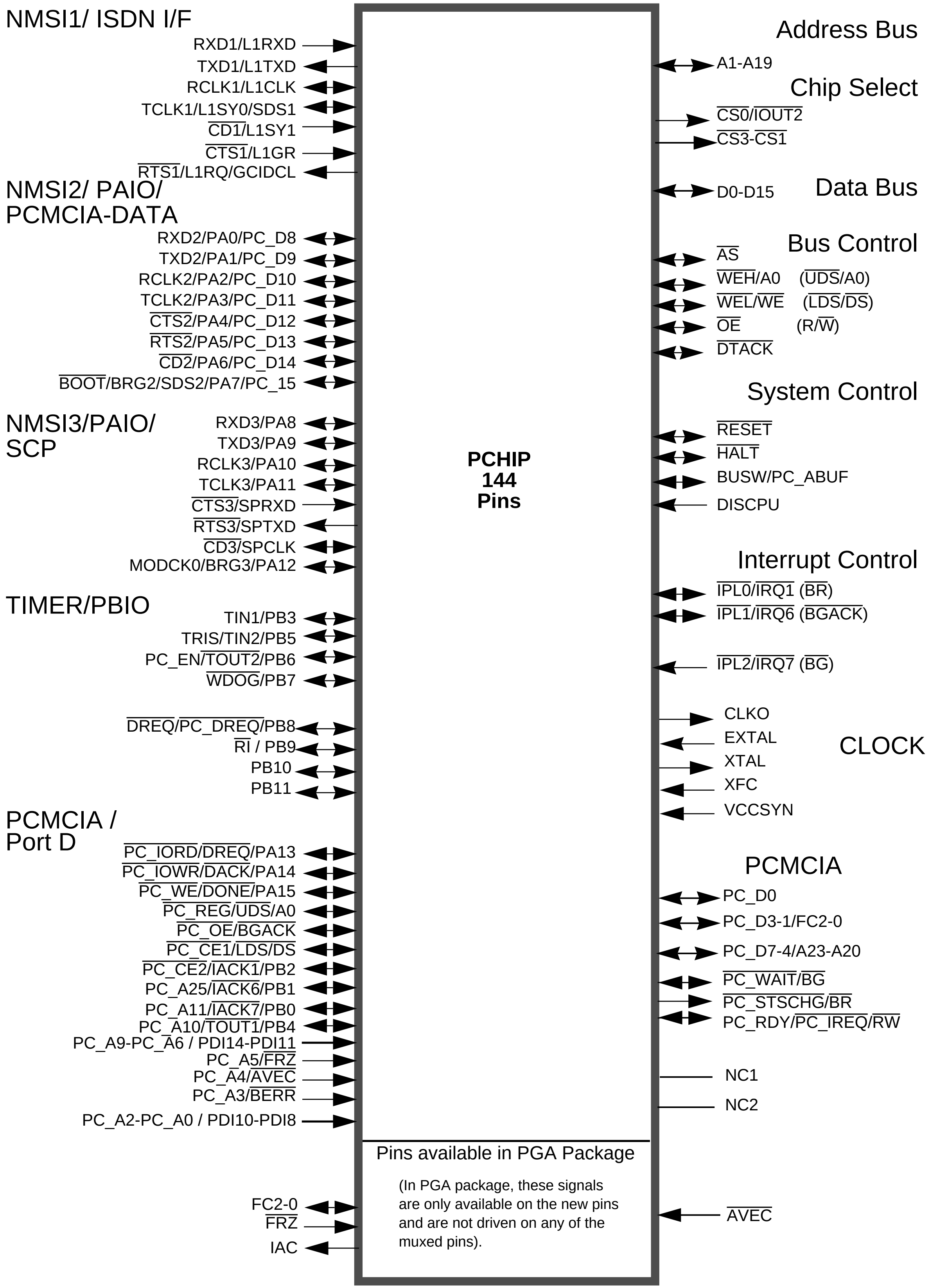
Table 6-1, Table 6-2, and Table 6-3 show the MC68PM302 signals in their functional groups. Each table includes a note describing in which modes (PCMCIA enable or Disable CPU) specific signals are available.

Table 6-13 defines the actual function of the pins as a function of the mode of operation.

The following paragraphs describe all the IMP signals and their functionality. The PCMCIA signals are described following those paragraphs. Please refer to Table 6-1, Table 6-2, and Table 6-3 for when a signal is available.

The PCMCIA interface is enabled by connecting the PC_EN pin high during reset. After reset, the PC_EN pin becomes $\overline{\text{TOUT2}}$/PB6. The CPU is disabled by connecting the DISCPU pin to VCC.

The IMP uses a M68000 like bus for communication between both on-chip and external peripherals. This bus is a single, continuous bus existing both on-chip and off-chip. Any access made internal to the device is visible externally. Any access made external is visible internally. Thus, when the M68000 core accesses the dual-port RAM, the bus signals are driven externally. Likewise, when an external device accesses an area of external system memory, the chip-select logic can be used to generate the chip-select signal and $\overline{\text{DTACK}}$.



Note: Signals in parenthesis () are only valid in a system where DISCPU = PC_EN = 1 (PCMCIA is enabled and the CPU is disabled).

Figure 6-1. Functional Signal Groups

Table 6-1. IMP System Bus Pins

Group	Signal Name	Mnemonic $^{\text{(See Note)}}$	I/O
Mode Pins	Threestate *	TRIS/TIN2/PB5	I/O#
	Bus Width *	BUSW/PC_ABUF	I/O#
	PCMICA Enable *	PC_EN/$\overline{\text{TOUT2}}$/PB6	I/O#
	Boot from SCC *	$\overline{\text{BOOT}}$/BRG2/SDS2/PA7	I/O#
	Disable CPU	DISCPU	I
	VCCSYN (Enable PLL)	VCCSYN	I
	Clock Mode Select*	MODCLK/BRG3/PA12	I/O#
Clock	Crystal Oscillator	EXTAL, XTAL	I O
	External Filter Capacitor	XFC	I
	Clock Out	CLKO	O
System Control	Reset	$\overline{\text{RESET}}$	I/O#
	Halt	$\overline{\text{HALT}}$	I/O
Address	Address Bus	A19-A1	I/O
Data	Data Bus 15-0	D15-D0	I/O
Bus Control	Address Strobe	$\overline{\text{AS}}$	I/O
	Output Enable (Read/Write)	OE (R/$\overline{\text{W}}$)	I/O
	Write Enable High/A0 (Upper Data Strobe / Address 0)	$\overline{\text{WEH}}$/A0 ($\overline{\text{UDS}}$/A0)	I/O
	Write Enable Low (Lower Data Strobe / Data Strobe)	$\overline{\text{WEL}}$/$\overline{\text{WE}}$ ($\overline{\text{LDS}}$/$\overline{\text{DS}}$)	I/O
	Data Transfer Acknowledge	$\overline{\text{DTACK}}$	I/O
Interrupt Control Bus Arbitration	Interrupt Priority Level 0/Int. Request 1(Bus Request)	$\overline{\text{IPL0}}$/$\overline{\text{IRQ1}}$ ($\overline{\text{BR}}$)	I/O
	Interrupt Priority Level 1/Int. Request 6(Bus Grant)	$\overline{\text{IPL1}}$/$\overline{\text{IRQ6}}$ ($\overline{\text{BGACK}}$)	I/O
	Interrupt Priority Level 2 /Int.Request 7(Bus Grant Acknowledge)	$\overline{\text{IPL2}}$/$\overline{\text{IRQ7}}$ ($\overline{\text{BG}}$)	I
Chip Select	Chip Select 0 / Interrupt Output 2	$\overline{\text{CS0}}$/$\overline{\text{IOUT2}}$	O
	Chip Select 1-3	$\overline{\text{CS1}}$-$\overline{\text{CS3}}$	O
Spare	Not Connected	NC1-2	
Power	Clock Synthesizer Ground	GNDSYN	I
	System Power Supply and Return	VCC, GND	I

Note: Signal Names in Parentheses are only valid when the CPU is Disabled and the PCMCIA is enabled (DISCPU=PC_EN=1). When the PCMCIA is disabled, these signals are available on the PCMCIA pins and these pins retain their primary functionality. When the CPU is enabled and the PCMCIA interface is enabled, these signals are not available.

# Input buffer has a Schmitt Trigger

* Pins sampled during RESET and change function after RESET

Table 6-2. Additional Pins in PGA Package

Group	Signal Name (see Note)	Mnemonic	I/O
	Internal Access	IAC	O
	Autovector	$\overline{\text{AVEC}}$	I
	Function Codes	FC2-0	I/O
	Freeze Pin for Timers	$\overline{\text{FRZ}}$	I#

Note: In the PGA package, these signals are only available on these additional pins. They will not be driven onto any other pins that they are muxed with.

# Input buffer has a Schmitt Trigger

Table 6-3. IMP Peripheral Pins

Group	Signal Name	Mnemonic (See Note)	I/O
NMSI1 or ISDN	Receive Data / Layer 1 Receive data	RXD1/L1RXD	I#
	Transmit Data / Layer 1 Transmit data	TXD1/L1TXD	O
	Receive Clock / Layer 1 clock	RCLK1/L1CLK	I/O#
	Transmit Clock / PCM Sync / Serial Data Strobe 1	TCLK1/L1SY0/SDS1	I/O#
	Carrier Detect / Layer 1 Sync	$\overline{\text{CD1}}$/L1SY1	I#
	Clear To Send / Layer 1 Grant	$\overline{\text{CTS1}}$/L1GR	I#
	Request To Send / Layer 1 Request / GCI Clock Out	$\overline{\text{RTS1}}$/L1RQ/GCIDCL	O
NMSI2 or PORT A or PCMCIA Data	Receive Data / Port A / **PCMCIA Data**	RXD2/PA0/**PC_D8**	I/O#
	Transmit Data / Port A / **PCMCIA Data**	TXD2/PA1/**PC_D9**	I/O#
	Receive Clock / Port A / **PCMCIA Data**	RCLK2/PA2/**PC_D10**	I/O#
	Transmit Clock / Port A / **PCMCIA Data**	TCLK2/PA3/**PC_D11**	I/O#
	Clear To Send / Port A / **PCMCIA Data**	$\overline{\text{CTS}}$2/PA4/P**C_D12**	I/O#
	Request To Send / Port A / **PCMCIA Data**	$\overline{\text{RTS2}}$/PA5/**PC_D13**	I/O#
	Carrier Detect / Port A / **PCMCIA Data**	$\overline{\text{CD2}}$/PA6/**PC_D14**	I/O#
	Baud Rate Generator 2 Output / Port A / **PCMCIA Data/**$\overline{\text{BOOT}}$	$\overline{\text{BOOT}}$/BRG2/PA7/ **PC_D15**	I/O#
NMSI3 or PORT A or SCP Pins	Receive Data / Port A	RXD3/PA8	I/O#
	Transmit Data / Port A	TXD3/PA9	I/O#
	Receive Clock / Port A	RCLK3/PA10	I/O#
	Transmit Clock / Port A	TCLK3/PA11	I/O#
	SCP Receive Serial Data / Clear To Send	SPRXD/$\overline{\text{CTS3}}$	I#
	SCP Transmit Serial Data / Request To Send	SPTXD/$\overline{\text{RTS3}}$	O
	SCP Clock / Carrier Detect	SPCLK/$\overline{\text{CD3}}$	I/O#
	Mode Clock Select/Baud Rate Generator 3 Output / Port A	MODCLK/BRG3/PA12/	I/O#
PCMCIA or IDMA Pins or PORT A	**PCMCIA IO Read** / DMA Request / Port A	$\overline{\textbf{PC\_IORD}}$/$\overline{\text{DREQ}}$/PA13	I/O
	PCMCIA IO Write / DMA Acknowledge / Port A	$\overline{\textbf{PC\_IOWR}}$/$\overline{\text{DACK}}$/PA14	I/O
	PCMCIA IO Write Enable / DMA Done / Port A	$\overline{\textbf{PC\_WE}}$/$\overline{\text{DONE}}$/PA15	I/O
PCMCIA or IACK or PORT B	**PCMCIA Address** / Interrupt Acknowledge / Port B	**PC_A11**/$\overline{\text{IACK7}}$/PB0	I/O#
	PCMCIA Address /Interrupt Acknowledge / Port B	**PC_A25**/$\overline{\text{IACK6}}$/PB1	I/O#
	PCMCIA Card Enable 2 / Interrupt Acknowledge / Port B	**PC_CE2**$\overline{\text{IACK1}}$/PB2	I/O#
PCMCIA or Arbitration	**PCMCIA Status Change** / Bus Request	$\overline{\textbf{PC\_STSCHG}}$/$\overline{\text{BR}}$	O
	PCMCIA Wait / Bus Grant	$\overline{\textbf{PC\_WAIT}}$/$\overline{\text{BG}}$	I/O
	PCMCIA Output Enable / Bus Grant Acknowledge	$\overline{\textbf{PC\_OE}}$/$\overline{\text{BGACk}}$	I/O
PCMCIA or 68k Bus	**PCMCIA Ready** / **PCMICA Interrupt Request** / Read Write	$\overline{\textbf{PC\_RDY}}$**/**$\overline{\textbf{PC\_IREQ}}$/R$\overline{\text{W}}$	I/O
	PCMCIA REG / Upper Data Strobe / Address 0	$\overline{\textbf{PC\_REG}}$/$\overline{\text{UDS}}$/A0	I/O
	PCMCIA Card Enable 1 / Lower Data Strobe / Data Strobe	$\overline{\textbf{PC\_CE1}}$/$\overline{\text{LDS}}$/$\overline{\text{DS}}$	I/O
	PCMCIA Address / Freeze	**PC_A5**/$\overline{\text{FRZ}}$	I#
	PCMCIA Address / Autovector	**PC_A4**/$\overline{\text{AVEC}}$	I#
	PCMCIA Address / Bus Error	**PC_A3**/$\overline{\text{BERR}}$	I#

Table 6-3. IMP Peripheral Pins

Group	Signal Name	Mnemonic (See Note)	I/O
PCMCIA or TIMER pins	Timer 1 Input / Port B	TIN1/PB3	I/O#
	PCMCIA Address / Timer 1 Output / Port B /	**PC_A10**/$\overline{\text{TOUT1}}$/PB4	I/O#
	Threestate/Timer 2 Input / Port B	TRIS/TIN2/PB5	I/O#
	PCMCIA Enable / Timer 2 Output / Port B /	**PC_EN**/$\overline{\text{TOUT2}}$/PB6	I/O#
	Watchdog / Port B	$\overline{\text{WDOG}}$/PB7	I/O#
PC_DMA	**PCMCIA DMA Request** / IDMA Request / Port B	$\overline{\text{DREQ}}$/**PC_DREQ**/PB8	I/O#
Port B I/O	Port B / Ring Indicate	PB9/$\overline{\text{RI}}$	I/O#
	Port B	PB10	I/O#
	Port B	PB11	I/O#
PCMCIA or IMP Signals	PCMCIA Low Data Bus	PC_D0	I/O#
	PCMCIA Low Data Bus / Function Codes	**PC_D3-1** / FC2-0	I/O#
	PCMCIA Low Data Bus / Address bus	**PC_D7-4** / A23-A20	I/O#
PCMCIA or Port D	**PCMCIA Address** / Port D	**PC_A9-A6**/PDI14-11	I#
	PCMCIA Address / Port D	**PC_A2-PC_A0**/PDI10-8	I#

Note: Signals in bold are enabled when the PCMCIA interface is enabled (PC_EN=1). If the PCMCIA interface is not enabled, then these pin are avaible as their secondary functions.
# Input buffer has a Schmitt Trigger

All pins except EXTAL and CLKO support TTL levels. EXTAL, when used as an input clock, requires CMOS levels. CLKO supplies a CMOS level output.

All outputs (except CLKO) can drive up to 100 pF. CLKO is designed to drive up to 50 pF.

6.1.1 Mode Pins

TRIS - Three State

This input is sampled during total system reset ($\overline{\text{RESET}}$ and $\overline{\text{HALT}}$ asserted). When asserted high, TRIS three-states all of the 68PM302 pins. This pin must be valid as long as $\overline{\text{RESET}}$ and $\overline{\text{HALT}}$ are asserted and have a hold time of 5ns after $\overline{\text{RESET}}$ and $\overline{\text{HALT}}$ are negated. After reset, this pin becomes TIN2/PB5.

BUSW—Bus Width Select

This input defines the M68000 processor mode (MC68000 or MC68008) and the data bus width (16 bits or 8 bits, respectively). This pin must be valid as long as $\overline{\text{RESET}}$ and $\overline{\text{HALT}}$ are asserted and have a hold time of 5ns after $\overline{\text{RESET}}$ and $\overline{\text{HALT}}$ are negated. After reset, this pin becomes PC_ABUF.

In 16-bit mode, all accesses to internal and external memory by the M68000 core, the IDMA, the SDMA, and any external masters may be 16 bits. In 8-bit mode, all M68000 core and IDMA accesses to internal and external memory are limited to 8 bits. Also in 8-bit mode, SDMA accesses to external memory are limited to 8 bits, but CP accesses to the CP side of the dual-port RAM continue to be 16 bits. In 8-bit mode, external accesses to internal memory are also limited to 8 bits at a time.

Low = 8-bit data bus, MC68008 core processor

High = 16-bit data bus, MC68000 core processor

NOTE

The system bus can be changed dynamically by changing the BSW and BSWEN bits in the PPR register (See 3.2 Programmable Data Bus Size Switch

PC_EN/$\overline{\text{TOUT2}}$/PB6

The PCMCIA interface is enabled by connecting the PC_EN pin high during reset (with 5ns hold time after negation of $\overline{\text{RESET}}$ and $\overline{\text{HALT}}$). After reset, the PC_EN pin becomes $\overline{\text{TOUT2}}$/PB6. If PC_EN is low during reset, the PM302 regains most of the functions of the MC68302.

$\overline{\text{BOOT}}$/SDS2/PA7/BRG2 - Boot From SCC

PA7 and PA5 pins are sampled at initialization to determine the boot mode. To enable Boot from SCC2 mode, PA7 has to be pulled LOW during Reset (with 5ns hold time after negation of $\overline{\text{RESET}}$ and $\overline{\text{HALT}}$).

If Boot mode is enabled, PA5 determines the Clock source to SCC2. This pin has to be valid for 100 clocks after the negation of $\overline{\text{RESET}}$ and $\overline{\text{HALT}}$. The user can pull it HIGH or LOW with an external resistor. If Boot mode is not enabled, then PA5 is not sampled at initialization (See 3.3 Load Boot Code from an SCC).

DISCPU—Disable CPU (M68000 Core)

The 68PM302 can be configured to work solely with an external CPU. In this mode the on-chip M68000 core CPU should be disabled by asserting the DISCPU pin high during a total system reset ($\overline{\text{RESET}}$ and $\overline{\text{HALT}}$ asserted). DISCPU may only be changed on a total system reset.

The DISCPU pin, for instance, allows use of several IMP family devices to provide more than three SCC channels without the need for bus isolation techniques. Only one of the IMP M68000 cores is active and services the other IMPs as peripherals (with their respective cores disabled). Refer to 3.8.2 Disable CPU Logic (M68000) for more details.

MODCLK/BRG3/PA12—Clock Mode Select

During reset, if VCCSYN is connected to VCC, the state of this input signal determines the frequency of the external clock that is to be used by the phase locked loop (PLL) in the clock synthesizer to generate the system clocks. This pin must be valid as long as $\overline{\text{RE-}}$

$\overline{\text{SET}}$ and $\overline{\text{HALT}}$ are asserted and have a hold time of 5ns after $\overline{\text{RESET}}$ and $\overline{\text{HALT}}$ are negated. After reset, this pin becomes BRG3/PA12.

Table 6-4 shows the default values of the PLL. When the PLL is disabled (VCCSYN=0), this pin functions as PA12. When the PLL is enabled (VCCSYN1), this pin is sampled as MODCLK at reset. After reset, MODCLK/PA12 is a general purpose I/O pin.

Table 6-4. Default Operation Mode of the PLL

VCCSYN	MODCLK	PLL	Multi. factor (MF+1)	EXTAL freq. (examples)	CLKIN to the PLL	PM302 System Clock
0	X	Disabled	x	-	=EXTAL	=EXTAL
1	0	Enabled	4	4.192MHz	4.192MHz	16.768 MHz
1	1	Enabled	401	32.768KHz	32.768KHz	13.14 MHz

VCCSYN—Analog PLL Circuit Power

This pin is dedicated to the PM302 analog PLL circuits and determines whether the PLL is enabled or not. When this pin is connected to Vcc, the PLL is enabled, and when this pin is connected to ground, the PLL is disabled. The voltage should be well regulated and the pin should be provided with an extremely low impedance path to the V_{CC} power rail. V_{CCSYN} should be bypassed to GND by a 0.1µF capacitor located as close as possible to the chip package.

6.1.2 Clock Pins

The clock pins are shown in Figure 6-2.

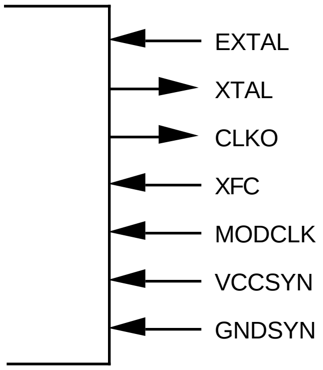


Figure 6-2. Clock Pins

EXTAL—External Clock/Crystal Input

This input provides two clock generation options (crystal and external clock). EXTAL may be used (with XTAL) to connect an external crystal to the on-chip oscillator and clock generator. If an external clock is used, the clock source should be connected to EXTAL, and

XTAL should be left unconnected. The oscillator uses an internal frequency equal to the external crystal frequency. The frequency of EXTAL may range from 0 Mhz to 20 MHz (or the maximum rated frequency, whichever is higher). When an external clock is used, it must provide a CMOS level at this input frequency.

In this manual, many references to the frequency “16.67 MHz” are made when the maximum operating frequency of the MC68PM302 is discussed. When using faster versions of the MC68PM302, such as 20 MHz, all references to 16.67 MHz may be replaced with 20 MHz. Note, however, that resulting parameters such as baud rates and timer periods change accordingly.

XTAL—Crystal Output

This output connects the on-chip oscillator output to an external crystal. If an external clock is used, XTAL should be left unconnected.

CLKO—Clock Out

This output clock signal is derived from the on-chip clock oscillator. This clock signal is internally connected to the clock input of the M68000 core, the communication processor, and system integration block. All M68000 bus timings are referenced to the CLKO signal. CLKO supports both CMOS and TTL output levels. The output drive capability of the CLKO signal is programmable to one-third, two-thirds, or full strength, or this output can be disabled.

XFC—IMP External Filter Capacitor

This pin is a connection for an external capacitor to filter the PLL.

GNDSYN—Analog PLL Circuits’ Ground

This pin is dedicated to the IMP analog PLL circuits. The pin should be provided with an extremely low impedance path to ground. GNDSYN should be bypassed to VCCSYN by a 0.1µF capacitor located as close as possible to the chip package.

6.1.3 System Control Pins

The system control pins are shown in Figure 6-3.

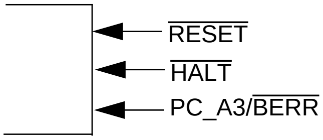


Figure 6-3. System Control Pins

$\overline{\text{RESET}}$ — Reset

This bidirectional, open-drain signal, acting as an input and asserted along with the $\overline{\text{HALT}}$ pin, starts an initialization sequence called a total system reset that resets the entire IMP.

$\overline{\text{RESET}}$ and $\overline{\text{HALT}}$ should remain asserted for at least 100 ms at power-on reset, and at least 10 clocks otherwise. The on-chip system RAM is not initialized during reset except for several locations initialized by the CP.

NOTE

With a 32.768KHz external crystal the minimum $\overline{\text{RESET}}$ length is 2.3 seconds

An internally generated reset, from the M68000 RESET instruction, causes the $\overline{\text{RESET}}$ line to become an output for 124 clocks. In this case, the M68000 core is not reset; however, the communication processor is fully reset, and the system integration block is almost fully reset (refer to 2.6 Internal Registers Map for a list of the unaffected registers). The user may also use the $\overline{\text{RESET}}$ output signal in this case to reset all external devices.

During a total system reset, the address, data, and bus control pins are all three-stated, except for $\overline{\text{CS3}}$–$\overline{\text{CS0}}$, $\overline{\text{WEH}}$, $\overline{\text{WEL}}$, and $\overline{\text{OE}}$, which are high, and IAC, which is low. The $\overline{\text{BG}}$ pin output is the same as that on the $\overline{\text{BR}}$ input. The general-purpose I/O pins are configured as inputs except for $\overline{\text{WDOG}}$, which is an open-drain output. The NMSI1 pins are all inputs, except for $\overline{\text{RTS1}}$ and TXD1, which output a high value. $\overline{\text{RTS3}}$ is high and CLKO is active.

NOTE

The $\overline{\text{RESET}}$ pin should not be asserted externally without also asserting the $\overline{\text{HALT}}$ pin. To reset just the internal IMP peripherals, the RESET instruction may be used.

Besides the total system reset and the RESET instruction, some of the IMP peripherals have reset bits in one of their registers that cause that particular peripheral to be reset to the same state as a total system reset or the RESET instruction. Reset bits may be found in the CP (in the CR), the IDMA (in the CMR), timer 1 (in the TMR1), and timer 2 (in the TMR2).

$\overline{\text{HALT}}$—Halt

When this bidirectional, open-drain signal is driven by an external device, it will cause the IMP bus master (M68000 core, SDMA, or IDMA) to stop at the completion of the current bus cycle. In PC_EN=1 DISCPU=0 mode, the external device can take ownership of the bus (See section 3.8.3 Bus Arbitration Logic). If the processor has stopped executing instructions due to a double-fault condition, this line is driven by the processor to indicate to external devices that the processor has stopped. An example of a double-fault condition is the occurrence of a bus error followed by a second bus error during the bus error exception stacking process. This signal is asserted with the $\overline{\text{RESET}}$ signal to cause a total IMP system reset. If $\overline{\text{BERR}}$ is asserted with the $\overline{\text{HALT}}$ signal, a retry cycle is performed.

PC_A3/$\overline{\text{BERR}}$–PCMCIA Address 3/Bus Error

When PC_EN=0 (PCMCIA is disabled), this pin functions as the input signal $\overline{\text{BERR}}$ which informs the bus master that there is a problem with the cycle currently being executed.

This signal is generated externally; however an internal $\overline{\text{BERR}}$ signal can be asserted by the on-chip hardware watchdog (bus timeout because of no $\overline{\text{DTACK}}$), by the chip-select logic (address conflict or write protect violation), or by external circuitry. If $\overline{\text{BERR}}$ is asserted with the $\overline{\text{HALT}}$ signal, a retry cycle is performed.

6.1.4 IMP Address Bus Pins (A23–A1)

The address bus pins are shown in Figure 6-4.

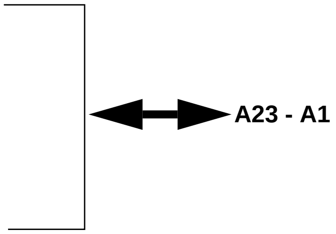


Figure 6-4. Address Bus Pins

A23—A1 form a 24-bit address bus when combined with $\overline{\text{WEH}}$/($\overline{\text{UDS}}$)/A0. The address bus is a bidirectional, three-state bus capable of addressing 16M bytes of data (including the IMP internal address space). It provides the address for bus operation during all cycles except CPU space cycles. In CPU space cycles, the CPU reads a peripheral device vector number.

These lines are outputs when the IMP (M68000 core, SDMA or IDMA) is the bus master and are inputs otherwise.

NOTE

When PCMCIA mode is enabled (pin PC_EN=1), A23-A20 signals are not available to the user. Internally they are driven by the current bus master, and the CS logic compares those signals. In this way, the address space is comprised of 4 banks of 1Mbyte. When an External master has ownership of the bus, A23-A20 are internally driven LOW.

NOTE

When PCMCIA mode is disabled PC_D7-PC_D4 become A23-A20.

6.1.5 IMP Data Bus Pins (D15—D0)

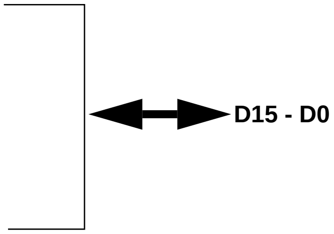


Figure 6-5. Data Bus Pins

This 16-bit, bidirectional, three-state bus is the general-purpose data path. It can transmit and accept data in either word or byte lengths. For all 16-bit IMP accesses, byte 0, the high-order byte of a word, is available on D15–D8, conforming to the standard M68000 format.

When working with an 8-bit bus (BUSW is low), the data is transferred through the low-order byte (D7–D0). The high-order byte (D15–D8) is not used for data transfer, but D8-D15 are outputs during write cycles and are not three-stated.

6.1.6 Bus Control Pins

The bus control pins are shown in Figure 6-6.

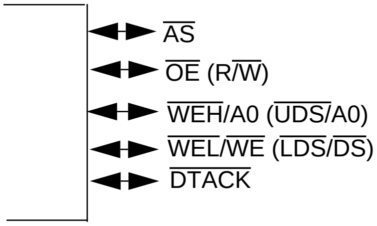

$\overline{AS}$
$\overline{OE}$ (R/$\overline{W}$)
$\overline{WEH}$/A0 ($\overline{UDS}$/A0)
$\overline{WEL}$/$\overline{WE}$ ($\overline{LDS}$/$\overline{DS}$)
$\overline{DTACK}$

Figure 6-6. Bus Control Pins

$\overline{\text{AS}}$—Address Strobe

This bidirectional signal indicates that there is a valid address on the address bus. This line is an output when the IMP (M68000 core, SDMA or IDMA) is the bus master and is an input otherwise.

$\overline{\text{OE}}$ (R/$\overline{\text{W}}$)— Output Enable (Read/Write)

This pin functions as R/$\overline{\text{W}}$ in Slave Mode and PCMCIA Enable mode (DISCPU=1 and PC_EN=1). In all other cases it functions as $\overline{\text{OE}}$.

As $\overline{\text{OE}}$, this output is active during a read cycle and indicates that an external device should place valid data on the bus.

As R/$\overline{\text{W}}$, this bidirectional signal defines the data bus transfer as a read or write cycle. It is an output when the PM302 is the bus master and is an input otherwise.

NOTE

This pin is high impedance when External Master acquires the Bus with $\overline{\text{HALT}}$.

$\overline{\text{WEH}}$ ($\overline{\text{UDS}}$/A0)—Write Enable High (Upper Data Strobe/Address 0)

This pin functions as $\overline{\text{UDS}}$ in Slave Mode and PCMCIA Enable (DISCPU=1 and PC_EN=1). In all other cases it functions as $\overline{\text{WEH}}$.

$\overline{\text{WEH}}$/A0 is active during a write cycle to indicate that an external device should expect data on the D15-D8 of the data bus. When using an 8-bit data bus this pin acts as A0.

$\overline{\text{UDS}}$ controls the flow of data on the data bus. When using a 16-bit data bus, this pin functions as upper data strobe ($\overline{\text{UDS}}$). When using an 8-bit data bus, this pin functions as A0. When used as A0 (i.e., the BUSW pin is low), the pin takes on the timing of the other address pins, as opposed to the strobe timing. This line is an output when the PM302 is the bus master and is an input otherwise.

NOTE

This pin is high impedance when External Master acquires the Bus with $\overline{\text{HALT}}$.

$\overline{\text{WEL}}$ ($\overline{\text{LDS/DS}}$)—Write Enable Low (Lower Data Strobe/Data Strobe)

This pin functions as $\overline{\text{LDS}}$ in Slave Mode and PCMCIA Enable (DISCPU=1 and PC_EN=1). In all other cases it functions as $\overline{\text{WEL}}$.

$\overline{\text{WEL}}$ is active during a write cycle to indicate that an external device should expect data on the D7-D0 of the data bus.

$\overline{\text{LDS}}$ controls the flow of data on the data bus. When using a 16-bit data bus, this pin functions as lower data strobe ($\overline{\text{LDS}}$). When using an 8-bit data bus, this pin functions as $\overline{\text{DS}}$. This line is an output when the PM302 (M68000 core, SDMA or IDMA) is the bus master and is an input otherwise.

NOTE

This pin is high impedance when External Master acquires the Bus with $\overline{\text{HALT}}$

DTACK—Data Transfer Acknowledge

This bidirectional signal indicates that the data transfer has been completed. DTACK can be generated internally in the chip-select logic either for an IMP bus master or for an external bus master access to an external address within the chip-select ranges. It will also be generated internally during any access to the on-chip dual-port RAM or internal registers. If DTACK is generated internally, then it is an output. It is an input when the IMP accesses an external device not within the range of the chip-select logic or when programmed to be generated externally.

6.1.7 Interrupt Control or Bus Arbitration Pins

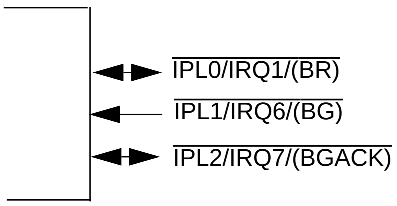


Figure 6-7. Interrupt Control and Bus Arbitration Pins

NOTE

These pins function as Arbitration only in Slave Mode and PCMCIA (DISCPU=1 and PC_EN=1).

IPL0/IRQ1/BR—Bus Request

The Bus Request pin is an open-drain output request signal from the IDMA and SDMA when the internal M68000 core is disabled.

As IPL2-IPL0 (normal mode), these input pins indicate the encoded priority level of the external device requesting an interrupt. Level 7 is the highest (nonmaskable) priority; while, level 0 indicates that no interrupt is requested. The least significant bit is IPL0, and the most significant bit is IPL2. These lines must remain stable until the M68000 core signals an interrupt acknowledged through FC2–FC0 and A19–A16 to ensure that the interrupt is properly recognized.

The IRQ1, IRQ6, and IRQ7 (dedicated mode) inputs indicate to the IMP that an external device is requesting an interrupt. Level 7 is the highest level and cannot be masked. Level 1 is the lowest level. Each one of these inputs (except for level 7) can be programmed to be either level-sensitive or edge-sensitive. The M68000 always treats a level 7 interrupt as edge sensitive.

NOTE

These pins function as bus arbitration pins only in Slave Mode and PCMCIA (DISCPU=1 and PC_EN=1).

Table 6-13 specifies in what mode each function is available.

IPL1/IRQ6/BG—Bus Grant

The Bus Grant pin is an input to the IDMA, SDMA and PCMCIA Controller when the internal M68000 core is disabled. During total system reset, BG = BR.

For Interrupt functionality see the description under Bus Request.

IPL2/IRQ7BGACK—Bus Grant Acknowledge

BGACK bidirectional signal indicates that some Master has taken ownership of the bus. This signal is an input when an external device owns the bus. This signal is an output when the IDMA, SDMA or PCMCIA Controller has become the master of the bus. If the SDMA steals a cycle from the IDMA, the BGACK pin will remain asserted continuously.

NOTE

BGACK should always be used in the external bus arbitration process. See 3.8.3 Bus Arbitration Logic for more details.

For Interrupt functionality see description under Bus Request.

6.1.8 Chip-Select Pins

The chip-select pins are shown in Figure 6-8.

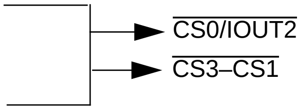


Figure 6-8. Chip-Select Pins

CS0/IOUT2—Chip-Select 0/Interrupt Output 2

In normal operation, this pin functions as CS0. CS0 is one of the four active-low output pins that function as chip selects for external devices or memory. It does not activate on accesses to the internal RAM or registers (including the BAR, SCR, or CKCR registers).

When the M68000 core is disabled, this pin operates as IOUT2. When the M68000 core is disabled IOUT2 provides the interrupt request output signal from the IMP interrupt controller to an external CPU. This signal is asserted if an internal Interrupt of level 4, 6, 7 is generated.

CS3–CS1—Chip Selects 3–1

These three active-low output pins function as chip selects for external devices or memory. CS3—CS0 do not activate on accesses to the internal RAM or registers (including the BAR, SCR, or CKCR registers).

6.1.9 No-Connect Pins

NC1–NC2 are reserved for future use and should not be connected.

6.1.10 PGA Package Pins

.

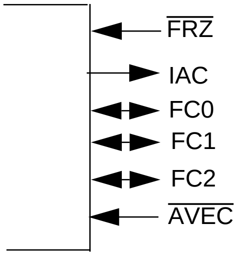


Figure 6-9. PGA Package Pins

$\overline{\text{FRZ}}$ - Freeze Activity

The $\overline{\text{FRZ}}$ pin is used to freeze the activity of selected peripherals. This is useful for systems debugging purposes. See section 3.1.5 Freeze Control for more details. $\overline{\text{FRZ}}$ should be continuously negated during total system reset.

IAC—Internal Access

The IAC signal is only available in the PGA package. This output indicates that the current bus cycle accesses an on-chip location. This includes the on-chip 4K byte block of internal RAM and registers (both real and reserved locations) and the system configuration registers ($0F0–$0FF). The above-mentioned bus cycle may originate from the M68000 core, the IDMA, or an external bus master. Note that, if the SDMA accesses the internal dual-port RAM, it does so without arbitration on the M68000 bus; therefore, the IAC pin is not asserted in this case. The timing of IAC is identical to that of the $\overline{\text{CS3}}$–$\overline{\text{CS0}}$ pins.

IFC2–FC0—Function Codes 2–0

These bidirectional signals indicate the state and the cycle type currently being executed. The information indicated by the function code outputs is valid whenever $\overline{\text{AS}}$ is active.

These lines are outputs when the IMP (M68000 core, SDMA, or IDMA) is the bus master and are inputs otherwise. The function codes output by the M68000 core are predefined; whereas, those output by the SDMA and IDMA are programmable. The function code lines are inputs to the chip-select logic and IMP internal register decoding in the BAR.

$\overline{\text{AVEC}}$— Autovector

When asserted during an interrupt acknowledge cycle, this pin indicates that the M68000 core should use automatic vectoring for an interrupt. This pin operates like $\overline{\text{VPA}}$ on the

MC68000, but is used for automatic vectoring only. $\overline{\text{AVEC}}$ instead of $\overline{\text{DTACK}}$ should be asserted during autovectoring and should be high otherwise

6.1.11 IMP Bus Interface Signal Summary

Table 6-5 and Table 6-5 summarize all bus signals discussed in the previous paragraphs. They show the direction of each pin for the following bus masters: M68000 core, IDMA, SDMA (includes DRAM refresh), PCMCIA Controller and External. Each bus master can access either internal dual-port RAM and registers or an external device or memory. When an external bus master accesses the internal dual-port RAM or registers, the access may be synchronous or asynchronous.

When the M68000 core is disabled, $\overline{\text{BR}}$ and $\overline{\text{BG}}$ change their direction.

Table 6-5. Bus Signal Summary—Core and External Master

		M68000 Core Master Access To		**External Master Access To**	
Signal Name	**Pin Type**	**Internal Memory Space**	**External Memory Space**	**Internal Memory Space**	**External Memory Space**
A23–A1, FC2–FC0, $\overline{\text{AS}}$, $\overline{\text{UDS}}$, $\overline{\text{LDS}}$, R/$\overline{\text{W}}$	I/O	O	O	I	I
$\overline{\text{WEH}}$,$\overline{\text{WEL}}$,$\overline{\text{OE}}$	O	O	O	O*	O*
IAC (available only in PGA)	O	O	O	O	O
D15–D0 Read	I/O	O	I	O	I
D15–D0 Write	I/O	O	O	I	I
$\overline{\text{DTACK}}$	I/O	O	**	O	**
$\overline{\text{BR}}$	I/O Open Drain	I	I	I	I
$\overline{\text{BG}}$	I/O	O	O	O	O
$\overline{\text{BGACK}}$	I/O	I	I	I	I
$\overline{\text{HALT}}$	I/O Open Drain	I/O	I/O	I	I
$\overline{\text{RESET}}$	I/O Open Drain	I/O	I/O	I	I
$\overline{\text{BERR}}$	I	I***	I***	I***	I***
$\overline{\text{IPL2–IPL0}}$	I	I	I	I	I
$\overline{\text{AVEC}}$	I	I	I	I	I
$\overline{\text{IOUT2}}$	O	O	O	O	O

* $\overline{\text{WEH}}$,$\overline{\text{WEL}}$,$\overline{\text{OE}}$ are high impedance when External Master Acquires the Bus with $\overline{\text{HALT}}$

** If $\overline{\text{DTACK}}$ is generated automatically (internally) by the chip-select logic, then it is an output. Otherwise, it is an input.

***$\overline{\text{BERR}}$ is input only in 68PM302; an internal $\overline{\text{BERR}}$ may be asserted by the IMP when the hardware watchdog is used or when the chip-select logic detects address conflict or write protect violation. $\overline{\text{BERR}}$ may be asserted by external logic in all cases.

6.1.12 Physical Layer Serial Interface Pins

The physical layer serial interface has 23 pins, and all of them have multiple functionality. The pins can be used in a variety of configurations in ISDN or non-ISDN environments. Table 6-6 shows the functionality of each group of pins and their internal connection to the three SCC and one SCP controller. The physical layer serial interface can be configured for non-multiplexed operation (NMSI) or multiplexed operation that includes IDL, GCI, and PCM highway modes. IDL and GCI are ISDN interfaces. When working in one of the multiplexed modes, the NMSI1/ISDN physical interface can be connected to all three SCC controllers.

Table 6-6. Serial Interface Pin Functions

First Function	Connected To	Second Function	Connected To
NMSI1 (8)	SCC1 Controller	ISDN Interface	SCC1/SCC2/SCC3
NMSI2 (8)	SCC2 Controller	PIO—Port A	Parallel I/O
NMSI3 (5)	SCC3 Controller	PIO—Port A	Parallel I/O
NMSI3 (3)	SCC3 Controller	SCP	SCP Controller

NOTE: Each one of the parallel I/O pins can be configured individually.

6.1.13 Typical Serial Interface Pin Configurations

Table 6-7 shows typical configurations of the physical layer interface pins for an ISDN environment. Table 6-8 shows potential configurations of the physical layer interface pins for a non-ISDN environment. The IDMA, IACK, and timer pins can be used in all applications either as dedicated functions or as PIO pins.

Table 6-7. Typical ISDN Configurations

Pins	Connected To	Used As
NMSI1 or ISDN I/F	SCC1 and SCC3	SCC1 Used as ISDN D-ch SCC3 Used as ISDN B2-ch
NMSI2	SCC2	SCC2 is Connected to Terminal
NMSI3	PA12–PA8	PIO (Extra Modem Signals and SCP Select Signals)
	SCP	Status/Control Exchange

NOTES:
1. ISDN environment with SCP port for status/control exchange and with existing terminal (for rate adaption).
2. D-ch is used for signaling.
3. B1-ch is used for voice (external CODEC required).
4. B2-ch is used for data transfer.

Table 6-8. Typical Generic Configurations

Pins	Connected To	Used As
NMSI1 or ISDN I/F	SCC1	Terminal with Modem
NMSI2	SCC2	Terminal with Modem
NMSI3 (5)	SCC3	Terminal without Modem
NMSI3 (3)	SCP	Status/Control Exchange

NOTE: Generic environment with three SCC ports (any protocol) and the SCP port. SCC3 does not use modem control signals.

6.1.14 NMSI1 or ISDN Interface Pins

The NMSI1 or ISDN interface pins are shown in Figure 6-10.

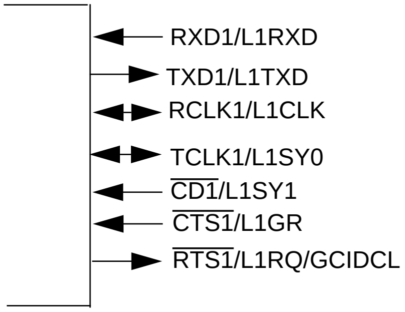

RXD1/L1RXD
TXD1/L1TXD
RCLK1/L1CLK
TCLK1/L1SY0
$\overline{\text{CD1}}$/L1SY1
$\overline{\text{CTS1}}$/L1GR
$\overline{\text{RTS1}}$/L1RQ/GCIDCL

Figure 6-10. NMSI1 or ISDN Interface Pins

These seven pins can be used either as NMSI1 in nonmultiplexed serial interface (NMSI) mode or as an ISDN physical layer interface in IDL, GCI, and PCM highway modes.

Table 6-9 shows the functionality of each pin in NMSI, GCI, IDL, and PCM highway modes.

Table 6-9. Mode Pin Functions

Signal Name	NMSI1		GCI		IDL		PCM	
RXD1/L1RXD	I	RXD1	I	L1RXD	I	L1RXD	I	L1RXD
TXD1/L1TXD	O	TXD1	O	L1TXD	O	L1TXD	O	L1TXD
RCLK1/L1CLK	I/O	RCLK1	I	L1CLK	I	L1CLK	I	L1CLK
TCLK1/L1SY0	I/O	TCLK1	O	SDS1	O	SDS1	I	L1SY0
$\overline{\text{CD1}}$/L1SY1	I	$\overline{\text{CD1}}$	I	L1SYNC	I	L1SYNC	I	L1SY1
$\overline{\text{CTS1}}$/L1GR	I	$\overline{\text{CTS1}}$	I	L1GR	I	L1GR		
$\overline{\text{RTS1}}$/L1RQ	O	$\overline{\text{RTS1}}$	O	GCIDCL	O	L1RQ	O	$\overline{\text{RTS}}$

NOTES:
1. In IDL and GCI mode, SDS2 is output on the PA7 pin.
2. $\overline{\text{CD1}}$ may be used as an external sync in NMSI mode.
3. $\overline{\text{RTS}}$ is the $\overline{\text{RTS1}}$, $\overline{\text{RTS2}}$, or $\overline{\text{RTS3}}$ pin according to which SCCs are connected to the PCM highway.

RXD1/L1RXD—Receive Data/Layer-1 Receive Data

This input is used as the NMSI1 receive data in NMSI mode and as the receive data input in IDL, GCI, and PCM modes.

TXD1/L1TXD—Transmit Data/Layer-1 Transmit Data

This output is used as NMSI1 transmit data in NMSI mode and as the transmit data output in IDL, GCI, and PCM modes. TXD1 may be configured as an open-drain output in NMSI mode. L1TXD in IDL and PCM mode is a three-state output. In GCI mode, it is an open-drain output.

RCLK1/L1CLK—Receive Clock/Layer-1 Clock

This pin is used as an NMSI1 bidirectional receive clock in NMSI mode or as an input clock in IDL, GCI, and PCM modes. In NMSI mode, this signal is an input when SCC1 is working with an external clock and is an output when SCC1 is working with its baud rate generator. The RCLK1 output can be three-stated by setting bit 14 in the DISC register.

TCLK1/L1SY0/SDS1—Transmit Clock/PCM Sync/Serial Data Strobe 1

This pin is used as an NMSI1 bidirectional transmit clock in NMSI mode, as a sync signal in PCM mode, or as the SDS1 output in IDL/GCI modes. In NMSI mode, this signal is an input when SCC1 is working with an external clock and is an output when SCC1 is working with its baud rate generator. The TCLK1 output can be three--stated by setting bit 15 in the DISC register.

NOTE

When using SCC1 in the NMSI mode with the internal baud rate generator operating, the TCLK1 and RCLK1 pins will always output the baud rate generator clock unless disabled in the CKCR register. Thus, if a dynamic selection between an internal and external clock source is required in an application, the clock pins should be disabled first in the CKCR register before switching the TCLK1 and RCLK1 lines. On SCC2 and SCC3, contention may be avoided by disabling the clock line outputs in the PACNT register.

In PCM mode, L1SY1–L1SY0 are encoded signals used to create channels that can be independently routed to the SCCs.

Table 6-10. PCM Mode Signals

L1SY1	L1SY0	Data (L1RXD, L1TXD) is Routed to SCC
0	0	L1TXD is Three-Stated, L1RXD is Ignored
0	1	CH-1
1	0	CH-2
1	1	CH-3

NOTE: CH-1, 2, and 3 are connected to the SCCs as determined in the SIMODE register.

In IDL/GCI modes, the SDS2–SDS1 outputs may be used to route the B1 and/or B2 channels to devices that do not support the IDL or GCI buses. This is configured in the serial interface mode (SIMODE) and serial interface mask (SIMASK) registers.

$\overline{\text{CD1}}$/L1SY1—Carrier Detect/Layer-1 Sync

This input is used as the NMSI1 carrier detect ($\overline{\text{CD}}$) pin in NMSI mode, as a PCM sync signal in PCM mode, and as an L1SYNC signal in IDL/GCI modes.

If the $\overline{\text{CD1}}$ pin has changed for more than one receive clock cycle, the IMP asserts the appropriate bit in the SCC1 event register. If the SCC1 channel is programmed not to support $\overline{\text{CD1}}$ automatically (in the SCC1 mode register), then this pin may be used as an external interrupt source. The current value of $\overline{\text{CD1}}$ may be read in the SCCS1 register. $\overline{\text{CD1}}$ may also be used as an external sync in NMSI mode. $\overline{\text{CD1}}$ will be grounded internally when this pin is used for an alternate function.

$\overline{\text{CTS1}}$/L1GR—Clear to Send/Layer-1 Grant

This input is the NMSI1 $\overline{\text{CTS}}$ signal in the NMSI mode or the grant signal in the IDL/GCI mode. If this pin is not used as a grant signal in GCI mode, it should be connected to V_{CC}.

If the $\overline{\text{CTS1}}$ pin has changed for more than one transmit clock cycle, the IMP asserts the appropriate bit in the SCC1 event register and optionally aborts the transmission of that frame.

If SCC1 is programmed not to support $\overline{\text{CTS1}}$ (in the SCC1 mode register), then this pin may be used as an external interrupt source. The current value of the $\overline{\text{CTS1}}$ pin may be read in the SCCS1 register.

$\overline{\text{RTS1}}$/L1RQ/GCIDCL—Request to Send/Layer-1 Request/GCI Clock Out

This output is the NMSI1 $\overline{\text{RTS}}$ signal in NMSI mode or PCM Highway mode, the IDL request signal in IDL mode, or the GCI data clock output in GCI mode. In PCM Highway mode, $\overline{\text{RTS1}}$ is asserted high.

$\overline{\text{RTS1}}$ is asserted when SCC1 (in NMSI mode) has data or pad (flags or syncs) to transmit.

In GCI mode this pin is used to output the GCI data clock.

6.1.15 NMSI2 Port or Port A Pins or PCMCIA Data Bus

The NMSI2 port or port A pins are shown in Figure 6-11. The PCMCIA function of these pins is shown in italics.

NOTE

> Those pins function as PAIO/NMSI2 when PCMCIA is disabled (PC_EN=0) or when PCMCIA is configured to 8 bit memory (16bit=0 in PCMR).

In non PCMCIA mode these eight pins can be used either as the NMSI2 port or as a general-purpose parallel I/O port. Each one of these pins can be configured individually to be general-purpose I/O pins or a dedicated function in NMSI2. When they are used as NMSI2 pins, they function exactly as the NMSI1 pins in NMSI mode.

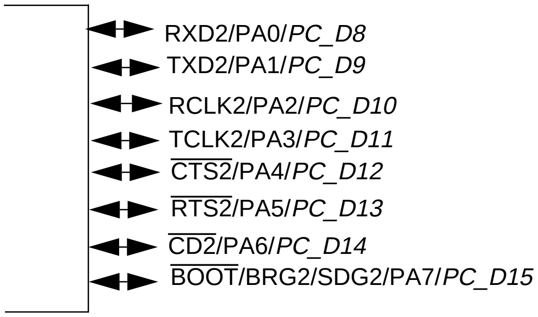

RXD2/PA0/*PC_D8*
TXD2/PA1/*PC_D9*
RCLK2/PA2/*PC_D10*
TCLK2/PA3/*PC_D11*
$\overline{CTS2}$/PA4/*PC_D12*
$\overline{RTS2}$/PA5/*PC_D13*
$\overline{CD2}$/PA6/*PC_D14*
$\overline{BOOT}$/BRG2/SDG2/PA7/*PC_D15*

Figure 6-11. NMSI2 Port or Port A Pins

The PA7 signal in dedicated mode becomes serial data strobe 2 (SDS2) in IDL and GCI modes. In IDL/GCI modes, the SDS2–SDS1 outputs may be used to route the B1 and/or B2 channels to devices that do not support the IDL or GCI buses. This is configured in the SIMODE and SIMASK registers. If SCC2 is in NMSI mode, this pin operates as BRG2, the output of the SCC2 baud rate generator, unless SDS2 is enabled to be asserted during the B1 or B2 channels of ISDN (bits SDC2–SDC1 of SIMODE). SDS2/BRG2 may be temporarily disabled by configuring it as a general-purpose output pin. TCLK2 acts as the SCC2 baud rate generator output if SCC2 is in one of the multiplexed modes.

- RXD2/PA0/*PC_D8*
- TXD2/PA1/*PC_D9*
- RCLK2/PA2/*PC_D10*
- TCLK2/PA3/*PC_D11*
- $\overline{CTS2}$/PA4/*PC_D12*
- $\overline{RTS2}$/PA5/*PC_D13*
- $\overline{CD2}$/PA6/*PC_D14*
- $\overline{BOOT}$/SDS2/PA7/BRG2/*PC_D15*

Table 6-11. Baud Rate Generator Outputs

Source	NMSI	GCI	IDL	PCM
SCC1				
SCC2	BRG2	TCLK2	TCLK2	TCLK2
SCC3	BRG3	TCLK3	TCLK3	TCLK3

NOTE: In NMSI mode, the baud rate generator outputs can also appear on the RCLK and TCLK pins as programmed in the SCON register.

PC_D15–PC_D8—PCMCIA Data Bus

The bidirectional PCMCIA card interfaces data bus. When the MC68PM302 is not enabled for PCMCIA, these pins are used for SCC2 signals.

6.1.16 NMSI3 Port or Port A Pins or SCP Pins

The NMSI3 port or port A pins or SCP pins are shown in Figure 6-12.

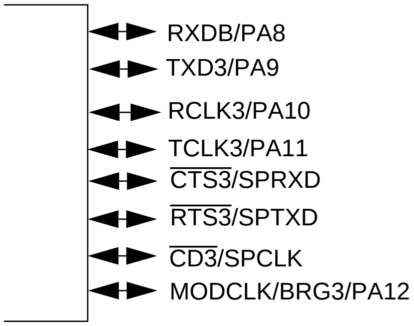

RXDB/PA8
TXD3/PA9
RCLK3/PA10
TCLK3/PA11
$\overline{\text{CTS3}}$/SPRXD
$\overline{\text{RTS3}}$/SPTXD
$\overline{\text{CD3}}$/SPCLK
MODCLK/BRG3/PA12

Figure 6-12. NMSI3 Port or Port A Pins or SCP Pins

These eight pins can be used either as the NMSI3 port or as the NMSI3 port (less three modem lines) and the SCP port. If the SCP is enabled (EN bit in SPMODE register is set), then the three lines are connected to the SCP port. Otherwise, they are connected to the SCC3 port.

Each of the port A I/O pins can be configured individually to be general-purpose I/O pins or a dedicated function in NMSI3. When they are used as the NMSI3 pins, they function exactly as the NMSI1 pins (see the previous description). TCLK3 acts as the SCC3 baud rate generator output if SCC3 is in one of the multiplexed modes.

- RXD3/PA8
- TXD3/PA9
- RCLK3/PA10
- TCLK3/PA11

SPRXD/$\overline{\text{CTS3}}$—SCP Receive Serial Data/NMSI3 Clear-to-Send Pin

This signal functions as the SCP receive data input or may be used as the NMSI3 $\overline{\text{CTS}}$ input pin.

SPTXD/$\overline{\text{RTS3}}$—SCP Transmit Serial Data/NMSI3 Request-to-Send Pin

This output is the SCP transmit data output or may be used as the NMSI3 $\overline{\text{RTS}}$ pin.

SPCLK/$\overline{\text{CD3}}$—SCP Clock/NMSI3 CD Pin

This bidirectional signal is used as the SCP clock output or the NMSI3 $\overline{\text{CD3}}$ input pin.

MODCLK/PA12/BRG3—Port A Bit 12/SCC3 Baud Rate Generator

This pin functions as bit 12 of port A or may be used as the SCC3 baud rate generator output clock when SCC3 is operating in NMSI mode.

NOTE

This pin is sampled at reset. Refer to section 6.1.1 for further details.

6.1.17 IDMA or Port A Pins or PCMCIA Pins

The IDMA or port A pins are shown in Figure 6-13.

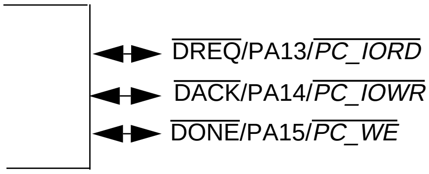

$\overline{\text{DREQ}}$/PA13/$\overline{\textit{PC\_IORD}}$
$\overline{\text{DACK}}$/PA14/$\overline{\textit{PC\_IOWR}}$
$\overline{\text{DONE}}$/PA15/$\overline{\textit{PC\_WE}}$

Figure 6-13. IDMA or Port A Pins

Each of these three pins can be used either as dedicated pins for the IDMA signals or as general-purpose parallel I/O port A pins. Note that even if one or more of the IDMA pins are used as general-purpose I/O pins, the IDMA can still be used. For example, if $\overline{\text{DONE}}$ is not needed by the IDMA, it can be configured as a general-purpose I/O pin. If the IDMA is used for memory-to-memory transfers only, then all three pins can be used as general-purpose I/O pins.

NOTE

These pins function as PCMCIA if PC_EN=1

$\overline{\text{DREQ}}$/PA13/PC_$\overline{\text{IORD}}$—DMA Request or Parallel IO or PCMCIA IO Read

DMA Request input is asserted by a peripheral device to request an operand transfer between that peripheral device and memory. In the cycle steal request generation mode, this input is edge-sensitive. In burst mode, it is level-sensitive.

PCMCIA I/O Read ($\overline{\text{IORD}}$) signal is made active to read data from the card's I/O space. The $\overline{\text{REG}}$ signal and at least one of $\overline{\text{CE1}}$ or $\overline{\text{CE2}}$ must also be active. A PC card will not respond to the $\overline{\text{IORD}}$ signal until it has been configured for I/O operation by the system.

$\overline{\text{DACK}}$/PA14/PC_$\overline{\text{IOWR}}$—DMA Acknowledge or Parallel IO or PCMCIA IO Write

DMA Acknowledge output, asserted by the IDMA, signals to the peripheral that an operand is being transferred in response to a previous transfer request.

PCMCIA IO Write ($\overline{\text{IOWR}}$) signal is made active to write data to the card's I/O space. The $\overline{\text{REG}}$ signal and at least one of $\overline{\text{CE1}}$ or $\overline{\text{CE2}}$ must also be active. A PC card will not respond to the $\overline{\text{IORD}}$ signal until it has been configured for I/O operation by the system.

DONE/PA15/PC_WE—DONE or Parallel IO or PCMCIA Write Enable/Program

DONE is a bidirectional, open-drain signal asserted by the IDMA or by a peripheral device during any IDMA bus cycle to indicate that the data being transferred is the last item in a block. The IDMA asserts this signal as an output during a bus cycle when the byte count register is decremented to zero. Otherwise, this pin is an input to the IDMA to terminate IDMA operation.

PCMCIA Write Enable/Program the WE/PGM input is used for strobing memory write data into the memory card.

6.1.18 IACK or PIO Port B Pins or PCMCIA

The IACK or PIO port B pins are shown in Figure 6-14.

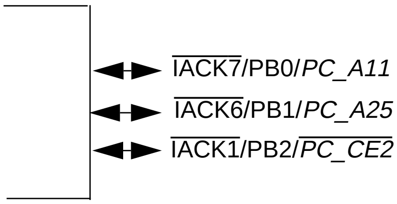


Figure 6-14. IACK or PIO Port B Pins

In non PCMCIA Mode each one of these three pins can be used either as an interrupt acknowledge signal or as a general-purpose parallel I/O port.

NOTE

These pins function as PCMCIA if PC_EN=1

NOTE

The IMP interrupt controller does not require the use of the IACK pins when it supplies the interrupt vector for the external source.

IACK7/PB0/PC_A11—Interrupt Acknowledge or Port B I/O or PCMCIA Address

IACK7 is an active low output signal that indicates to the external device that the IMP is executing an interrupt acknowledge cycle. The external device must then place its vector number on the lower byte of the data bus or use AVEC for autovectoring (unless internal vector generation is used).

The PC_A11 input is driven by the PCMCIA card interface. (See Section 5 PCMCIA Controller)

IIACK6/PB1/PC_A25—Interrupt Acknowledge or Port B I/O or PCMCIA Address

IACK6 is an active low output signal that indicates to the external device that the IMP is executing an interrupt acknowledge cycle. The external device must then place its vector number on the lower byte of the data bus or use AVEC for autovectoring (unless internal vector generation is used).

The PC_A25 (PCMCIA card interface) input address line is used together with the $\overline{\text{PC\_REG}}$ signal to identify different memory regions (See Section 5 PCMCIA Controller).

$\overline{\text{IACK1}}$/PB2/$\overline{\text{PC\_CE2}}$—Interrupt Acknowledge or Port B I/O or PCMCIA Control

$\overline{\text{IACK1}}$ is an active low output signal that indicates to the external device that the IMP is executing an interrupt acknowledge cycle. The external device must then place its vector number on the lower byte of the data bus or use **$\overline{\text{AVEC}}$** for autovectoring (unless internal vector generation is used).

The $\overline{\text{PC\_CE2}}$ interfaces to the PCMCIA Connector, and is asserted by the PC Host when it selects the Card.

6.1.19 Timer Pins

The timer pins are shown in Figure 6-15.

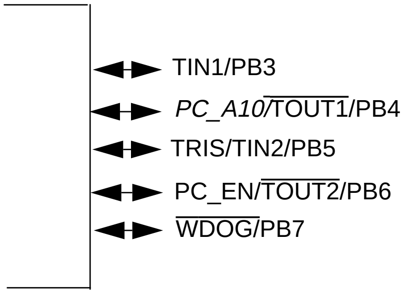

TIN1/PB3
PC_A10/$\overline{\text{TOUT1}}$/PB4
TRIS/TIN2/PB5
PC_EN/$\overline{\text{TOUT2}}$/PB6
$\overline{\text{WDOG}}$/PB7

Figure 6-15. Timer Pins

Each of these five pins can be used either as a dedicated timer function or as a general-purpose port B I/O port pin. Note that the timers do not require the use of external pins.

TIN1/PB3—Timer 1 Input or Parallel IO

This input is used as a timer clock source for timer 1 or as a trigger for the timer 1 capture register. TIN1 may also be used as the external clock source for any or all three SCC baud rate generators.

$\overline{\text{TOUT1}}$/PB4/PC_A10—Timer 1 Output or Parallel IO or PCMCIA Address

The $\overline{\text{TOUT1}}$ output is used as an active-low pulse timeout or an event overflow output (toggle) from timer 1.

The PC_A10 input is driven by the PCMCIA card interface. (See Section 5 PCMCIA Controller)

TIN2/PB5—Timer 2 Input or Parallel IO

This input can be used as a timer clock source for timer 2 or as a trigger for the timer 2 capture register.

TOUT2/PB6—Timer 2 Output

This output is used as an active-low pulse timeout or as an event overflow output (toggle) from timer 2.

WDOG/PB7—Watchdog Output

This active-low, open-drain output indicates expiration of the watchdog timer. WDOG may be externally connected to the RESET and HALT pins to reset the IMP. When WDOG is asserted, it will remain asserted for 16 IMP system clock cycles (if the IMP PLL is not enabled by the MODCLK pins), or 16 clock cycles after the PLL locks (if the PLL is enabled by the MODCLK pins). WDOG is never asserted by the on-chip hardware watchdog (see the BERR signal description). The WDOG pin function is enabled after a total system reset. It may be reassigned as the PB7 I/O pin in the PBCNT register.

6.1.20 Parallel I/O Pins with Interrupt Capability

The four parallel I/O pins with interrupt are shown in Figure 6-16.

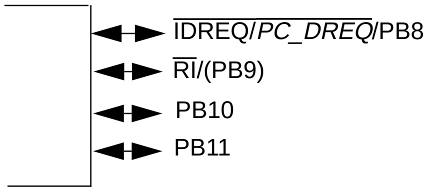


Figure 6-16. Port B Parallel I/O Pins with Interrupt

PB11 and PB8—Port B Parallel I/O Pins

These four pins may be configured as a general-purpose parallel I/O ports with interrupt capability. Each of the pins can be configured either as an input or an output. When configured as an input, each pin can generate a separate, maskable interrupt on a high-to-low transition. PB8 may also be used to request a refresh cycle from the DRAM refresh controller rather than as an I/O pin.

IDREQ/PC_DREQ/PB8

If the PC_DMA function is enabled, this pin is an output and functions as PC_DREQ. The user can connect it to SPKR, INPACK or IOIS16 pins on the PCMCIA Connector (see PC-MCIA Draft on DMA). If the PC_DMA function is not enabled, this pin may be configured as a general-purpose parallel I/O port with interrupt capability (if enabled in the Interrupt Controller) or as DREQ input to the IDMA (if enabled in the IDMA). The user should enable either the Interrupt or IDMA. This pin can be configured either as an input or an output. When configured as an input, this pin can generate a separate, maskable interrupt on a high-to-low transition. Refer to section 5.3.11 for further details on this pin muxing.

NOTE

> It is possible to enable Wake-Up from Low Power when PIT is expired (which replaces PB8 interrupt) and have $\overline{\text{DREQ}}$ or $\overline{\text{PC\_DREQ}}$ at the same system.

PB9(RI)

This pin may be configured as a general-purpose parallel I/O port with interrupt capability and wake-up capability.This pin can be configured either as an input or an output. When configured as an input, this pin can generate a separate, maskable interrupt on a high-to-low transition.

This pin can also wake-up the IMP from low power STOP, DOZE, or STAND-BY mode when a high to low transition occurs on this pin when configured as an input. See 2.4.4.2.3 IMP Wake-Up from Low Power STOP Modes for more information.

In addition, when the MC68PM302 is configured for the PCMCIA interface, this pin can be connected internally to the PCMCIA I/O Event Indication Register or directly to the PC_$\overline{\text{STSCHG}}$ pin. For more details on this feature refer to 5.1.9 PCMCIA Ring Indication.

PB10

This pin may be configured as a general-purpose parallel I/O port with interrupt capability and wake-up capability.This pin can be configured either as an input or an output. When configured as an input, this pin can generate a separate, maskable interrupt on a high-to-low transition.

This pin can also wake-up the IMP from low power STOP, DOZE, or STAND-BY mode when a high to low transition occurs on this pin when configured as an input. See 2.4.4.2.3 IMP Wake-Up from Low Power STOP Modes for more information.

6.1.21 PCMCIA or Port D Pins

The PCMCIA or port D pins are shown in Figure 6-17. When the PCMCIA is disabled, 7 of the PCMCIA pins become input only port pins. The PCMCIA function of these pins is shown in italics. Refer to 6.1.23 IMP Bus Control or PCMCIA Pins for PCMCIA pin description.

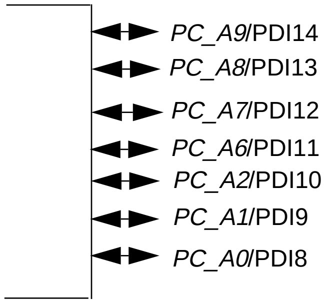


Figure 6-17. Port D Pins

6.1.22 IMP Arbitration or PCMCIA Pins

NOTE

These pins take their PCMCIA function if PC_EN=1

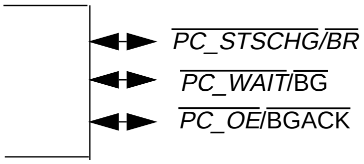

$\overline{\text{PC\_STSCHG}}$/$\overline{\text{BR}}$
$\overline{\text{PC\_WAIT}}$/$\overline{\text{BG}}$
$\overline{\text{PC\_OE}}$/$\overline{\text{BGACK}}$

Figure 6-18. PCMCIA Signals

PC_$\overline{\text{WAIT}}$/$\overline{\text{BG}}$—PCMCIA Extend Bus Cycle or Bus Request

The Bus Grant (in Enable CPU Mode) output indicates to all external bus masters devices that the processor will release the bus control at the end of the current bus cycle. In Slave Mode, this signal is an input to the IDMA, SDMA or PCMCIA Controller. During total system reset $\overline{\text{BR}}$=$\overline{\text{BG}}$**.**

PC Wait, the $\overline{\text{WAIT}}$ signal, is asserted by the MC68PM302 to delay the memory space accesses or I/O space accesses.

PC_$\overline{\text{STSCHG}}$/$\overline{\text{BR}}$—PCMCIA Status Changed [replace BVD1] or Bus Grant

The Bus Request (in Enable CPU Mode) input indicates to the on-chip bus arbiter that an external device desires to become Bus Master. In Slave Mode this signal is an open-drain output request from the IDMA, SDMA or PCMCIA Controller.

$\overline{\text{STSCHG}}$ is an optional signal used to alert the system to changes in the ready/busy (RDY/$\overline{\text{BSY}}$), write protect (WP), or battery voltage (BV) conditions of the card while the I/O interface is configured.

This signal is inactive (high) when this function is not supported by the card or when the SigChg bit in the card status register is false (logic 0). When the SigChg bit is true (logic

1) this signal is active when the changed bit in the card status register is true (logic 1). The changed bit is logical OR of the individual changed bits - CBVD1, CBVD2, CWP, and CRDBSY - in the pin replacement register.

PC_OE/BGACK—PCMCIA Output Enable or Bus Grant Acknowledge

The BGACK bidirectional signal indicates that some other device besides the M68000 core has become the bus master. This signal is an input when an external device or the M68000 core owns the bus. This signal is an output when the IDMA, SDMA or the PCMCIA Controller has become master of the bus. If the SDMA steals a cycle from the IDMA, the BGACK pin will remain asserted continuously.

PC Output Enable (PC_OE line) is an active low, input signal used to gate memory read data from the memory card.

6.1.23 IMP Bus Control or PCMCIA Pins

These pins are muxed and therefore replace other pins described in the previous paragraphs. The replaced signal is indicated in each description.

NOTE

These pins take their PCMCIA function if PC_EN=1

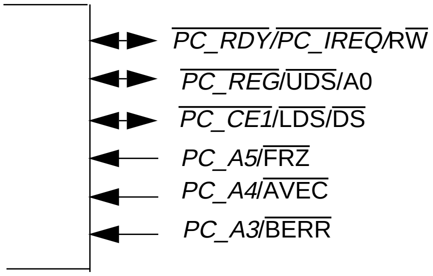


Figure 6-19. PCMCIA Signals or IMP Bus Control

PC_RDY/PC_IREQ/RW—PCMCIA Ready/Busy/PCMCIA Interrupt Request or 68K Read/Write

In PCMCIA Mode the Ready/Busy (RDY/BSY) function is provided by this signal only when the card and the host are configured for the memory-only interface. When a host socket and the card inserted into it are both configured for the I/O interface, this function is provided by the RDY/BSY status bit in the card's pin replacement register.

This signal is available as PCMCIA Interrupt Request (PC_IREQ) only when the card and the interface are configured for the I/O and memory interface. The interrupt request is level only. This pin serves as RDY/BSY in memory-only cards, and as PC_IREQ for I/O cards. This pin is used as RDY/$\overline{\text{BSY}}$ while an I/O capable card is configured for the memory-only interface.

In Non-PCMCIA Mode this pin functions as R/$\overline{\text{W}}$ (see section 6.1.6 Bus Control Pins).

PC_$\overline{\text{REG}}$/$\overline{\text{UDS}}$/A0—PCMCIA Attribute Memory Select or Upper Data Strobe/Address

In PCMCIA Mode this input is kept inactive (high) for all common memory accesses. When this signal is active, access is limited to attribute memory ($\overline{\text{OE}}$ or $\overline{\text{WE}}$ active) and to the I/O space ($\overline{\text{IORD}}$ or $\overline{\text{IOWR}}$ active).

In Non-PCMCIA Mode this pin functions as $\overline{\text{UDS}}$/A0 (see section 6.1.6 Bus Control Pins).

PC_$\overline{\text{CE1}}$/$\overline{\text{LDS}}$—PCMCIA Card Enable or Lower Data Strobe

The $\overline{\text{PC\_CE2}}$ pin interfaces to the PCMCIA Connector, and is asserted by the PC Host when it selects the Card.

In Non-PCMCIA Mode this pin functions as $\overline{\text{LDS}}$ (see section 6.1.6 Bus Control Pins).

PC_A5/$\overline{\text{FRZ}}$—PCMCIA Address or Freeze Activity

The PC_A5 input is driven by the PCMCIA card interface. (See Section 5 PCMCIA Controller)

The $\overline{\text{FRZ}}$ pin is used to freeze the activity of selected peripherals. This is useful for systems debugging purposes. $\overline{\text{FRZ}}$ should be continuously negated during total system reset (See section 3.6.6 for more information about this function).

PC_A4/$\overline{\text{AVEC}}$—PCMCIA Address or Autovector

The PC_A4 input is driven by the PCMCIA card interface. (See Section 5 PCMCIA Controller)

$\overline{\text{AVEC}}$, when asserted during an interrupt acknowledge cycle, indicates that the M68000 core should use automatic vectoring for an interrupt. This pin operates like $\overline{\text{VPA}}$ on the MC68000, but is used for automatic vectoring only. $\overline{\text{AVEC}}$ instead of $\overline{\text{DTACK}}$ should be asserted during autovectoring and should be high otherwise.

PC_A3/$\overline{\text{BERR}}$—PCMCIA Address or Bus Error

The PC_A4 input is driven by the PCMCIA card interface. (See Section 5 PCMCIA Controller.)

The Bus Error input signal informs the bus master (M68000 core, SDMA, IDMA, or external bus master) that there is a problem with the cycle currently being executed. The on-chip hardware watchdog (bus timeout because of no $\overline{\text{DTACK}}$) and the chip-select logic (address conflict or write-protect violation), can assert the internal bus error signal to the current bus master. If $\overline{\text{BERR}}$ is asserted with the $\overline{\text{HALT}}$ signal, a retry cycle is performed.

6.1.24 IMP Address and Function Codes or PCMCIA Data

NOTE

These pins take their PCMCIA function if PC_EN=1.

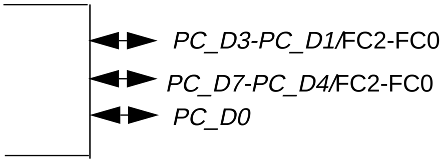


Figure 6-20. PCMCIA Data Bus or IMP Address/FC

PC_D15–PC_D0/FC2-FC0—PCMCIA Data Bus or Function Codes

In PCMCIA mode this bidirectional Data Bus interfaces to the Card Connector.

As FC2-FC0, these bidirectional signals indicate the state and the cycle type currently being executed. The information indicated by the function code outputs is valid whenever $\overline{\text{AS}}$ is active.

These lines are outputs when the IMP (M68000 core, SDMA,IDMA or PCMCIA Controller) is the bus master and are inputs otherwise. The function codes output by the M68000 core are predefined; whereas, those output by the SDMA and IDMA are programmable. The function code lines are inputs to the chip-select logic and IMP internal register decoding in the BAR.The bidirectional PCMCIA card interfaces to the data bus. When the MC68PM302 is not enabled for PCMCIA, these pins are used for SCC2 signals, function codes, and additional address lines A23-A20.

6.2 SUMMARY OF PIN MULTIPLEXING

The following table defines the actual function of the pins as a function of the mode of operation. On the left column the pin is identified with all its possible functions. In the table cells the actual function of the pin is specified according to the mode defined in the column header. This table specifies pin muxing determined by mode pins PC_EN and DISCPU and by the Package type (TQFP or PGA); it does not specify muxing determined by software programming of registers.

Table 6-12. Pin Muxing

Pin Name	PC_EN=1 DISCPU=0 TQFP	PC_EN=1 DISCPU=0 PGA	PC_EN=1 DISCPU=1 TQFP	PC_EN=1 DISCPU=1 PGA	PC_EN=0 DISCPU=0 TQFP	PC_EN=0 DISCPU=0 PGA	PC_EN=0 DISCPU=1 TQFP	PC_EN=0 DISCPU=1 PGA
PC_D7-PC_D4/A23-A20	PC_D7-PC_D4	PC_D7-PC_D4	PC_D7-PC_D4	PC_D7-PC_D4	A23-A20	A23-A20	A23-A20	A23-A20
PC_D3-PC_D1/FC2-FC0	PC_D3-PC_D1	PC_D3-PC_D1	PC_D3-PC_D1	PC_D3-PC_D1	FC2-FC0	-	FC2-FC0	-
PC_D15-PC_D8/ NMSI2/PA7-PA0	If 16bit=0 in PCMR NMSI2/PA7-PA0	If 16bit=0 in PCMR NMSI2/PA7-PA0	If 16bit=0 in PCMR NMSI2/PA7-PA0	If 16bit=0 in PCMR NMSI2/PA7-PA0	NMSI2/PA7-PA0	NMSI2/PA7-PA0	NMSI2/PA7-PA0	NMSI2/PA7-PA0
	If 16bit=1 in PCMR PC_D15-PC_D8	If 16bit=1 in PCMR PC_D15-PC_D8	If 16bit=1 in PCMR PC_D15-PC_D8	If 16bit=1 in PCMR PC_D15-PC_D8				
PC_WAIT/BG	PC_WAIT	PC_WAIT	PC_WAIT	PC_WAIT	BG	BG	BG	BG
PC_STSCHG/BR	PC_STSCHG	PC_STSCHG	PC_STSCHG	PC_STSCHG	BR	BR	BR	BR
PC_RDY/RW	PC_RDY	PC_RDY	PC_RDY	PC_RDY	RW	RW	RW	RW
PC_IOWR/DACK/PA14	PC_IOWR	PC_IOWR	PC_IOWR	PC_IOWR	DACK/PA14	DACK/PA14	DACK/PA14	DACK/PA14
PC_WE/DONE/PA15	PC_WE	PC_WE	PC_WE	PC_WE	DONE/PA15	DONE/PA15	DONE/PA15	DONE/PA15
PC_REG/UDS/A0	PC_REG	PC_REG	PC_REG	PC_REG	UDS/A0	UDS/A0	UDS/A0	UDS/A0
PC_OE/BGACK	PC_OE	PC_OE	PC_OE	PC_OE	BGACK	BGACK	BGACK	BGACK
PC_CE1/LDS/DS	PC_CE1	PC_CE1	PC_CE1	PC_CE1	LDS/DS	LDS/DS	LDS/DS	LDS/DS
PC_CE2/IACK1/PB2	PC_CE2	PC_CE2	PC_CE2	PC_CE2	IACK1/PB2	IACK1/PB2	IACK1/PB2	IACK1/PB2
PC_A25/IACK6/PB1	PC_A25	PC_A25	PC_A25	PC_A25	/IACK6/PB1	/IACK6/PB1	/IACK6/PB1	/IACK6/PB1
PC_A11/IACK7/PB0	PC_A11	PC_A11	PC_A11	PC_A11	IACK7/PB0	IACK7/PB0	IACK7/PB0	IACK7/PB0
PC_A10/TOUT1/PB4	PC_A10	PC_A10	PC_A10	PC_A10	TOUT1/PB4	TOUT1/PB4	TOUT1/PB4	TOUT1/PB4
PC_A9-PC_A6/PDI14-PDI11	PC_A9-PC_A6	PC_A9-PC_A6	PC_A9-PC_A6	PC_A9-PC_A6	PDI14-PDI11	PDI14-PDI11	PDI14-PDI11	PDI14-PDI11
PC_A5/FRZ	PC_A5	PC_A5	PC_A5	PC_A5	FRZ	-	FRZ	-
PC_A4/AVEC	PC_A4	PC_A4	PC_A4	PC_A4	AVEC	-	AVEC	-
PC_A3/BERR	PC_A3	PC_A3	PC_A3	PC_A3	BERR	BERR	BERR	BERR
PC_A2-PC_A0/PDI10-PDI8	PC_A2-PC_A0	PC_A2-PC_A0	PC_A2-PC_A0	PC_A2-PC_A0	PDI10-PDI8	PDI10-PDI8	PDI10-PDI8	PDI10-PDI8
PC_DREQ/DREQ/PB8	PC_DREQ/ DREQ/PB8	PC_DREQ/ DREQ/PB8	PC_DREQ/ DREQ/PB8	PC_DREQ/ DREQ/PB8	PB8	PB8	PB8	PB8
PC_IORD/DREQ/PA13	PC_IORD	PC_IORD	PC_IORD	PC_IORD	DREQ/PA13	DREQ/PA13	DREQ/PA13	DREQ/PA13
IPL0/IRQ1/BR	IPL0/IRQ1	IPL0/IRQ1	BR	BR	IPL0/IRQ1	IPL0/IRQ1	IPL0/IRQ1	IPL0/IRQ1
IPL1/IRQ6/BGACK	IPL1/IRQ6	IPL1/IRQ6	BGACK	BGACK	IPL1/IRQ6	IPL1/IRQ6	IPL1/IRQ6	IPL1/IRQ6
IPL2/IRQ7/BG	IPL2/IRQ7	IPL2/IRQ7	BG	BG	IPL2/IRQ7	IPL2/IRQ7	IPL2/IRQ7	IPL2/IRQ7
WEH/UDS/A0	WEH/A0	WEH/A0	UDS/A0	UDS/A0	WEH*	WEH*	WEH*	WEH*
WEL/LDS/DS	WEL	WEL	LDS/DS	LDS/DS	WEL	WEL	WEL	WEL
OE/RW	OE	OE	RW	RW	OE	OE	OE	OE
FC2-FC0 (PGA)	-	FC2-FC0	-	FC2-FC0	-	FC2-FC0	-	FC2-FC0
FRZ (PGA)	-	FRZ	-	FRZ	-	FRZ	-	FRZ
AVEC (PGA)	-	AVEC	-	AVEC	-	AVEC	-	AVEC
IAC (PGA)	-	IAC	-	IAC	-	IAC	-	IAC

*This pin is not used in 8-bit mode

6.3 POWER AND GROUND PINS

The MC68PM302 has power supply pins throughout the outer four rows/columns and V_{CC} pins on the V_{CC} ring located on the perimeter of the inner ground pins. Special attention should be paid to connecting the IMP PLL power pins designated by VCCSYN and GNDSYN respectively (see 2.4.3.6 IMP PLL Pins). Careful attention has been paid to reducing IMP noise, potential cross-talk, and RF radiation from the output drivers. Inputs may be +5 V when V_{CC} is 0 V without damaging the device.

6.4 WHEN TO USE PULLUP RESISTORS

Pins that are input-only or output-only do not require external pullups. The bidirectional bus control signals require pullups since they are three-stated by the processor when they are not being driven. Open-drain signals always require pullups.

Unused inputs should not be left floating. If they are input-only, they may be tied directly to V_{CC} or ground, or a pullup or pulldown resistor may be used. Unused outputs may be left unconnected. Unused I/O pins may be configured as outputs after reset and left unconnected.

If the IMP is to be held in reset for extended periods of time in an application (other than what occurs in normal power-on reset or board test sequences) due to a special application requirement (such as V_{CC} dropping below required specifications, etc.), then three-stated signals and inputs should be pulled up or down. This decreases stress on the device transistors and saves power.

See the $\overline{\text{RESET}}$ pin description for the condition of all pins during reset.

SECTION 7
ELECTRICAL CHARACTERISTICS

The AC specifications presented consist of output delays, input setup and hold times, and signal skew times. All signals are specified relative to an appropriate edge of the clock (CLKO pin) and possibly to one or more other signals.

VERY IMPORTANT NOTE REGARDING SIGNALS

> A few signals have been added to and removed from the 68PM302 or their functionality has changed. Several signals are only available when 68302 is in CPU disable mode or when the PCMCIA interface is disabled. The IAC signal is only available on the PGA package. The RMC and BCLR signals have been removed (please ignore any reference to these two signals).The following diagrams and tables show the timing for all available signals. For complete information on which signals are available in which modes (CPU disable and/or PCMCIA disable), please refer to Section 6 Signal Description.

7.1 MAXIMUM RATINGS

Rating	Symbol	Value	Unit	
Supply Voltage	V_{DD}	– 0.3 to + 7.0	V	This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to his high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either GND or V_{DD})
Input Voltage	V_{in}	– 0.3 to + 7.0	V	
Current Drain per Pin		excluding V_{CC} and V_{SS}	mA	
Operating Temperature Range MC68PM302 MC68PM302C	T_A	 0 to 70 – 40 to 85	°C	
Storage Temperature Range	T_{stg}	– 55 to + 150	°C	

7.2 THERMAL CHARACTERISTICS

$T_J = T_A + (P_D \bullet \theta_{JA})$

$P_D = (V_{DD} \bullet I_{DD}) + P_{I/O}$

where:

$P_{I/O}$ is the power dissipation on pins.

7.3 POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$T_J = T_A + (P_D \bullet q_{JA})(1)$

where:

T_A=Ambient Temperature, ∞C

q_{JA}=Package Thermal Resistance, Junction to Ambient, ∞C/W

P_D=$P_{INT} + P_{I/O}$

P_{INT}=I_{DD} x V_{DD}, Watts—Chip Internal Power

$P_{I/O}$=Power Dissipation on Input and Output Pins—User Determined

For most applications $P_{I/O} < 0.3 \bullet P_{INT}$ and can be neglected. If $P_{I/O}$ is neglected, an approximate relationship between P_D and T_J is

$P_D = K \prod (T_J + 273\infty C)(2)$

Solving equations (1) and (2) for K gives:

$K = P_D \bullet (T_A + 273\infty C) + q_{JA} \bullet P_D^2(3)$

where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

7.4 POWER DISSIPATION (SEE NOTE)

Characteristic Power Dissipation	Symbol	5 Volt		Unit
		Typ	Max	
Normal Mode at 20 MHz	$P_{D(I)}$	70	TBD	mA
Normal Mode at 16 MHz	$P_{D(I)}$	60	TBD	mA
Low Power Standby Mode	$P_{DSB(I)}$	7	TBD	mA
Low Power Doze Mode	$P_{DDZ(I)}$	500	TBD	µA
Low Power Stop Mode	$P_{DSP(I)}$	100	TBD	µA

Note: These values are preliminary estimates. Test values are TBD.

7.4.1 Layout Practices

Each V_{CC} pin on the MC68PM302 should be provided with a low-impedance path to the board's supply. Each GND pin should likewise be provided with a low-impedance path to ground. The power supply pins drive distinct groups of logic on chip. The V_{CC} power supply should be bypassed to ground using at least four 0.1 µF by-pass capacitors located as close as possible to the four sides of the package. The capacitor leads and associated printed circuit traces connecting to chip V_{CC} and GND should be kept to less than 1/2" per capacitor lead. A four-layer board is recommended, employing two inner layers as V_{CC} and GND planes. All output pins on the MC68PM302 have fast rise and fall times. Printed circuit (PC) trace interconnection length should be minimized in order to minimize undershoot and reflections caused by these fast output switching times. Maximum PC trace lengths on the order of 6" are recommended. Capacitance calculations should consider all device loads as well as parasitic capacitances due to the PC traces. Attention to proper PCB layout and bypassing becomes especially critical in systems with higher capacitive loads because these loads create higher transient currents in the V_{CC} and GND circuits. Pull up all unused inputs or signals that will be inputs during reset. Special care should be taken to minimize the noise levels on the PLL supply pins.

7.4.2 Power Dissipation Considerations.

Power dissipation is a key issue in portable applications. Some of the factors which affect current consumption are described in this section. Most of the current consumed by CMOS devices is alternating current (AC) which is charging and discharging the capacitances of the pins and internal nodes. Therefore, the total current consumption is the sum of these internal and external currents.

This current consumption is described by the formula:

$I=CVf$
whereC=node/pin capacitance
V=voltage swing and
f=frequency of node/pin toggle.

Example: For a pin loaded with a 50pF capacitance, operating at 5.5V and with a toggling rate of 10MHz, the current consumption is:

$I=50 \bullet 10^{-12} \bullet 5.5 \bullet 10 \bullet 10^{6} = 2.75mA$

The maximum internal current value (I_{CCI}-max), reflects the maximum possible switching of the internal buses, which is not necessarily a real application case. The typical internal current value ($I_{CCI\text{-}typ}$) reflects the average switching of the internal buses.

For applications which require very low current consumption it is recommended to:

a) minimize external memory accesses, and use internal memory accesses instead;
b) minimize the number of pins which are switching;
c) minimize the capacitive load on the pins;
d) connect the unused inputs to pull-up or pull-down resistors.

7.4.3 DC Electrical Characteristics

Characteristic	Symbol	Min	Max	Unit
Input High Voltage (Except pins noted below)	V_{IH}	2.0	V_{DD}	V
Input High Voltage(PC_A2-PC_A11, PC_A25,PC_CE2, PC_D7-PC_D0, CTS1, CD1, RXD1, TCLK1, BRG1, RCLK1, RTS1, RXD2, TXD2, RCLK2, TCLK2, CTS2, RTS2, CD2, BRG2, RXD3, CD3, RCLK3, TCLK3, PA12, RTS3, TXD3, CTS3, TOUT2, TOUT1, TIN2, TIN1, WDOG, PB8-11,PC_ABUF,FRZ_) (These pins have schmitt trigger inputs)	V_{IH}	2.5	V_{DD}	V
Input Low Voltage	V_{IL}	V_{SS} – 0.3	0.8	V
Input Undershoot Voltage	V_{CIL}	-	-0.8	V
Input High Voltage (EXTAL) 3.3 Volt or 5 Volt Part	V_{CIH}	.8 * V_{DD}	V_{DD}	V
Input High Voltage (RESET)	V_{IHR}	2.5	V_{DD}	V
Input Leakage Current:	II_N	—	20	µA
Input Capacitance All Pins	C_{IN}	—	15	pF
Three-State Leakage Current (2.4/0.5 V)	I_{TSI}	– 20	20	µA
Open Drain Leakage Current (2.4 V)	I_{OD}	—	20	µA
Output High Voltage (I_{OH} = 400 µA) (see Note)	V_{OH}	2.4	—	V
Output Low Voltage	V_{OL}			V
(I_{OL} = 3.2 mA) A1–A23, PB0–PB11, FC0–FC2, CS0–CS3, IACK, BG, RCLK1, RCLK2, RCLK3, TCLK1, TCLK2, TCLK3, RTS1, RTS2, RTS3, SDS2, PA12, RXD2, RXD3, CTS2, CD2, CD3, DREQ, BRG1		—	0.5	
(I_{OL} = 5.3 mA) AS, UDS, LDS, R/W, BERR BGACK, DTACK, DACK, D0–D15, RESET		—	0.5	
(I_{OL} = 7.0 mA) TXD1, TXD2, TXD3		—	0.5	
(I_{OL} = 8.9 mA) DONE, HALT, BR (as output)		—	0.5	
(I_{OL} = 3.2 mA) CLKO		—	0.4	
(I_{OL} = 3.2 mA) All other pins		—	0.5	
Output Drive CLKO Output Drive Except CLKO)	O_{CLK} O_{ALL}	— —	50 100	pF pF
Output Drive Derating Factor for CLKO of 0.030 ns/pF Output Drive Derating Factor for CLKO of 0.025 ns/pF Output Drive Derating Factor for All Other Pins 0.025 ns/pF Output Drive Derating Factor for All Other Pins 0.05 ns/pF	O_{KF} O_{KF} O_{KF} O_{KF}	20 50 20 100	50 130 100 200	pF pF pF pF
Power 5.0 Volt Part 3.3 Volt Part	V_{DD}	4.5 3.0	5.5 3.6	V
Common	V_{SS}	0	0	V

NOTE:The maximum I_{OH} for a given pin is one-half the I_{OL} rating for that pin. For an I_{OH} between 400 µA and I_{OL}/2 mA, the minimum V_{OH} is calculated as: V_{DD} – (1 +.05 V/mA(I_{OH} –.400 mA)).

7.5 DC ELECTRICAL CHARACTERISTICS—NMSI1 IN IDL MODE

Characteristic	Symbol	Min	Max	Unit	Condition
Input Pin Characteristics: L1CLK, L1SY1, L1RXD, L1GR					
Input Low Level Voltage	V_{IL}	-10%	+ 20%	V	(% of V_{DD})
Input High Level Voltage	V_{IH}	V_{DD} - 20%	V_{DD} + 10%	V	
Input Low Level Current	II_L	—	± 10	μA	$V_{in} = V_{ss}$
Input High Level Current	II_H	—	± 10	μA	$V_{in} = V_{DD}$
Output Pin Characteristics: L1TXD, SDS1- SDS2, L1RQ					
Output Low Level Voltage	V_{OL}	0	1.0	V	I_{OL} = 5.0 mA
Output High Level Voltage	V_{OH}	V_{DD} - 1.0	V_{DD}	V	I_{OH} = 400 μA

7.6 IMP CHARACTERISTICS

7.6.1 IMP AC ELECTRICAL SPECIFICATIONS CONTROL TIMING

(GND = 0 Vdc, TA = 0 to 70°C; The electrical specifications in this document are preliminary; See Figure 7-1)

			16.67 MHz		20 MHz		25 MHz		
Num.	**Characteristic**	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Unit**
	System Frequency	fsys	dc1	16.67	dc1	20.00	dc	25.00	MHz
	Crystal Frequency	f_{XTAL}	25	6000	25	6000	25	6000	kHz
	On-Chip VCO System Frequency	fsys	10	16.67	10	20	10	25	MHz
	Start-up Time With external clock (oscillator disabled) or after changing the multiplication factor MF. With external crystal oscillator enabled.	t_{pll} t_{osc}		2500 75,000		2500 75,000		2500 75,000	clks
	CLKO stability	Δ CLK	TBD	TBD	TBD	TBD	TBD	TBD	%
1	CLKO Period	t_{cyc}	60	—	50	—	40	-	ns
1A	EXTAL Duty Cycle	t_{dcyc}	40	60	40	60	40	60	%
1C	External Clock Input Period	t_{EXTcyc}	60	—	50	—	40	-	ns
2, 3	CLKO Pulse Width (measured at 1.5v)	t_{CW1}	TBD	—	TBD	—	TBD	-	ns
4, 5	CLKO Rise and Fall Times (full drive)	t_{Crf1}	—	5	—	4	-	4	ns
5B	EXTAL to CLKO skew (PLL Disabled)	t_{EXTP1}	2	11	2	9	2	7	ns
5C, 5D	EXTAL Rise and Fall Times (PLL Disabled)	t_{EXTRF}	—	5	—	5	—	5	ns

Note: The minimum VCO frequency and the PLL default values put some restrictions on the minimum system frequency.

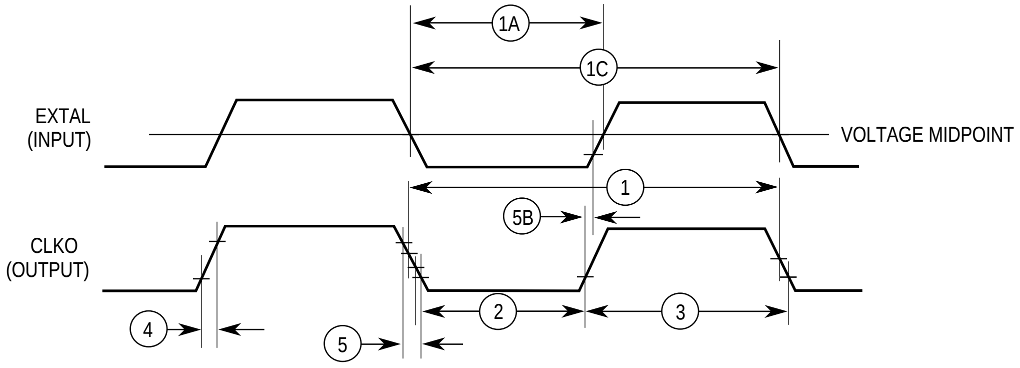


Figure 7-1. Clock Timing

7.6.2 AC Electrical Characteristics - IMP Phased Lock Loop (PLL) Characteristics

Characteristics	Expression	Min	Max	Unit
VCO frequency when PLL enabled	MF * Ef	10	f (Note 1.)	MHz
PLL external capacitor (XFC pin to VCCSYN)	MF * C_{XFC} (Note 1.) @ MF ≤ 5 @ MF > 5	 MF * 340 MF * 380	 MF * 480 MF * 970	pF

[1] f is the maximum operating frequency. Ef is EXTAL frequency. CXFC is the value of the PLL capacitor (connected between XFC pin and VCCSYN) for MF=1. The recommended value for C_{XFC} is 400pF for MF ≤ 5 and 540pF for MF> 5. The maximum VCO frequency is limited to the internal operation frequency, as defined above.

Examples:

1. VCCSYN = MODCK = 1

2. Crystal is 32.768 KHz (or 4.192 MHz), initial MF = 401, initial frequency = 13.14 MHz; later, MF is changed to 762 to support a frequency of 25 MHz.

Minimum c_{XFC} is: 762 x 380 = 289 nF, maximum c_{XFC} is: 401 x 970 = 390 nF. The recommended c_{XFC} for 25 MHz is: 762 x 540 = 414 nF.

289 nF < c_{XFC} < 390 nF and closer to 414 nF. The proper available value for c_{XFC} is 390 nF.

3. In order to get higher range, higher crystal frequency can be used (i.e. 50 KHz), in this case:

Minimum c_{XFC} is: 667 x 380 = 253 nF. Maximum c_{XFC} is: 401 x 970 = 390 nF. Therefore 253 nF < c_{XFC} < 390 nF.

7.6.3 IMP DC Electrical Characteristics—NMSI1 in IDL Mode

Characteristic	Symbol	Min	Max	Unit	Condition
Input Pin Characteristics: L1CLK, L1SY1, L1RXD, L1GR					
Input Low Level Voltage	V_{IL}	–10%	+ 20%	V	(% of V_{DD})
Input High Level Voltage	V_{IH}	V_{DD} – 20%	V_{DD} + 10%	V	
Input Low Level Current	II_L	—	± 10	µA	$V_{in} = V_{ss}$
Input High Level Current	II_H	—	± 10	µA	$V_{in} = V_{DD}$
Output Pin Characteristics: L1TXD, SDS1– SDS2, L1RQ					
Output Low Level Voltage	V_{OL}	0	1.0	V	I_{OL} = 5.0 mA
Output High Level Voltage	V_{OH}	V_{DD} – 1.0	V_{DD}	V	I_{OH} = 400 µA

7.6.4 AC Electrical Specifications—IMP Bus Master Cycles

(see Figure 7-2, Figure 7-3, Figure 7-4, and Figure 7-5))

Num.	Characteristic	Symbol	16.67 MHz @5.0 V		20 MHz @5.0 V		25 MHz @5.0 V		Unit
			Min	Max	Min	Max	Min	Max	
6	Clock High to FC, Address Valid	$t_{CHFCADV}$	0	45	0	40	0	30	ns
7	Clock High to Address, Data Bus High Impedance (Maximum)	t_{CHADZ}	—	50	—	42	—	33	ns
8	Clock High to Address, FC Invalid (Minimum)	t_{CHAFI}	0	—	0	—	0	—	ns
9	Clock High to $\overline{AS}$, $\overline{DS}$ Asserted (see Note 1)	t_{CHSL}	3	30	3	25	3	20	ns
11	Address, FC Valid to $\overline{AS}$, $\overline{DS}$ Asserted (Read) $\overline{AS}$ Asserted Write (see Note 2)	t_{AFCVSL}	15	—	12	—	10	—	ns
12	Clock Low to $\overline{AS}$, $\overline{DS}$ Negated (see Note 1)	t_{CLSH}	—	30	—	25	—	20	ns
13	$\overline{AS}$, $\overline{DS}$ Negated to Address, FC Invalid (see Note 2)	t_{SHAFI}	15	—	12	—	10	—	ns
14	$\overline{AS}$ (and $\overline{DS}$ Read) Width Asserted (see Note 2)	t_{SL}	120	—	100	—	80	—	ns
14A	$\overline{DS}$ Width Asserted, Write (see Note 2)	t_{DSL}	60	—	50	—	40	—	ns
15	$\overline{AS}$, $\overline{DS}$ Width Negated (see Note 2)	t_{SH}	60	—	50	—	40	—	ns
16	Clock High to Control Bus High Impedance	t_{CHCZ}	—	50	—	42	—	33	ns
17	$\overline{AS}$, $\overline{DS}$ Negated to R/$\overline{W}$ Invalid (see Note 2)	t_{SHRH}	15	—	12	—	10	—	ns
18	Clock High to R/$\overline{W}$ High (see Note 1)	t_{CHRH}	—	30	—	25	—	20	ns
20	Clock High to R/$\overline{W}$ Low (see Note 1)	t_{CHRL}	—	30	—	25	—	20	ns
20A	$\overline{AS}$ Asserted to R/$\overline{W}$ Low (Write) (see Notes 2 and 6)	t_{ASRV}	—	10	—	10	—	7	ns
21	Address FC Valid to R/$\overline{W}$ Low (Write) (see Note 2)	t_{AFCVRL}	15	—	12	—	10	—	ns
22	R/$\overline{W}$ Low to $\overline{DS}$ Asserted (Write) (see Note 2)	t_{RLSL}	30	—	25	—	20	—	ns
23	Clock Low to Data-Out Valid	t_{CLDO}	—	30	—	25	—	20	ns
25	$\overline{AS}$, $\overline{DS}$, Negated to Data-Out Invalid (Write) (see Note 2)	t_{SHDOI}	15	—	12	—	10	—	ns
26	Data-Out Valid to $\overline{DS}$ Asserted (Write) (see Note 2)	t_{DOSL}	15	—	12	—	10	—	ns
27	Data-In Valid to Clock Low (Setup Time on Read) (see Note 5)	t_{DICL}	7	—	6	—	5	—	ns
28	$\overline{AS}$, $\overline{DS}$ Negated to $\overline{DTACK}$ Negated (Asynchronous Hold) (see Note 2)	t_{SHDAH}	0	110	0	95	0	75	ns
29	$\overline{AS}$, $\overline{DS}$ Negated to Data-In Invalid (Hold Time on Read)	t_{SHDII}	0	—	0	—	—	—	ns
30	$\overline{AS}$, $\overline{DS}$ Negated to $\overline{BERR}$ Negated	t_{SHBEH}	0	—	0	—	0	—	ns
31	$\overline{DTACK}$ Asserted to Data-In Valid (Setup Time) (see Notes 2 and 5)	t_{DALDI}	—	50	—	42	—	33	ns
32	$\overline{HALT}$ and $\overline{RESET}$ Input Transition Time	t_{RHr}, t_{RHf}	—	150	—	150	—	150	ns
33	Clock High to $\overline{BG}$ Asserted	t_{CHGL}	—	30	—	25	—	20	ns
34	Clock High to $\overline{BG}$ Negated	t_{CHGH}	—	30	—	25	—	20	ns
35	$\overline{BR}$ Asserted to $\overline{BG}$ Asserted (see Note 11)	t_{BRLGL}	2.5	4.5	2.5	4.5	2.5	4.5	clks
36	$\overline{BR}$ Negated to $\overline{BG}$ Negated (see Note 7)	t_{BRHGH}	1.5	2.5	1.5	2.5	1.5	2.5	clks

Num.	Characteristic	Symbol	16.67 MHz		20 MHz		25 MHz		Unit
			Min	Max	Min	Max	Min	Max	
37	$\overline{\text{BGACK}}$ Asserted to $\overline{\text{BG}}$ Negated	t_{GALGH}	2.5	4.5	2.5	4.5	2.5	4.5	clks
37A	$\overline{\text{BGACK}}$ Asserted to $\overline{\text{BR}}$ Negated (see Note 8)	t_{GALBRH}	10	1.5	10	1.5	10	1.5	ns/ clks
38	$\overline{\text{BG}}$ Asserted to Control, Address, Data Bus High Impedance ($\overline{\text{AS}}$ Negated)	t_{GLZ}	—	50	—	42	—	33	ns
39	$\overline{\text{BG}}$ Width Negated	t_{GH}	1.5	—	1.5	—	1.5	—	clks
40	$\overline{\text{BGACK}}$ Asserted to Address Valid	t_{GALAV}	15	—	15	—	15	—	ns
41	$\overline{\text{BGACK}}$ Asserted to $\overline{\text{AS}}$ Asserted	t_{GALASA}	30	—	30	—	20	—	ns
44	$\overline{\text{AS}}$, $\overline{\text{DS}}$ Negated to $\overline{\text{AVEC}}$ Negated	t_{SHVPH}	0	50	0	42	0	33	ns
46	$\overline{\text{BGACK}}$ Width Low	t_{GAL}	1.5	—	1.5	—	1.5	—	clks
47	Asynchronous Input Setup Time (see Note 5)	t_{ASI}	10	—	10	—	7	—	ns
48	$\overline{\text{BERR}}$ Asserted to $\overline{\text{DTACK}}$ Asserted (see Notes 2 and 3)	t_{BELDAL}	10	—	10	—	7	—	ns
53	Data-Out Hold from Clock High	t_{CHDOI}	0	—	0	—	0	—	ns
55	R/$\overline{\text{W}}$ Asserted to Data Bus Impedance Change	t_{RLDBD}	0	—	0	—	0	—	ns
56	$\overline{\text{HALT}}$/$\overline{\text{RESET}}$ Pulse Width (see Note 4)	t_{HRPW}	10	—	10	—	10	—	clks
57	$\overline{\text{BGACK}}$ Negated to $\overline{\text{AS}}$, $\overline{\text{DS}}$, R/$\overline{\text{W}}$ Driven	t_{GASD}	1.5	—	1.5	—	1.5	—	clks
57A	$\overline{\text{BGACK}}$ Negated to FC	t_{GAFD}	1	—	1	—	1	—	clks
58	$\overline{\text{BR}}$ Negated to $\overline{\text{AS}}$, $\overline{\text{DS}}$, R/$\overline{\text{W}}$ Driven (see Note 7)	t_{RHSD}	1.5	—	1.5	—	1.5	—	clks
58A	$\overline{\text{BR}}$ Negated to FC (see Note 7)	t_{RHFD}	1	—	1	—	1	—	clks

NOTES:

1. For loading capacitance of less than or equal to 50 pF, subtract 4 ns from the value given in the maximum columns.
2. Actual value depends on clock period since signals are driven/latched on different CLKO edges. To calculate the actual spec for other clock frequencies, the user may derive the formula for each specification. First, derive the margin factor as:

   M = N(P/2) – Sa

   where N is the number of one-half CLKO periods between the two events as derived from the timing diagram,

   P is the rated clock period of the device for which the specs were derived (e.g., 40 ns with a 25-MHz device), and Sa is the actual spec in the data sheet. Thus, for spec 14 at 25 MHz:

   M = 5(40 ns/2) – 80 ns = 20 ns.

   Once the margin (M) is calculated for a given spec, a new value of that spec (Sn) at another clock frequency with period (Pa) is calculated as:

   Sn = N(Pa/2) – M

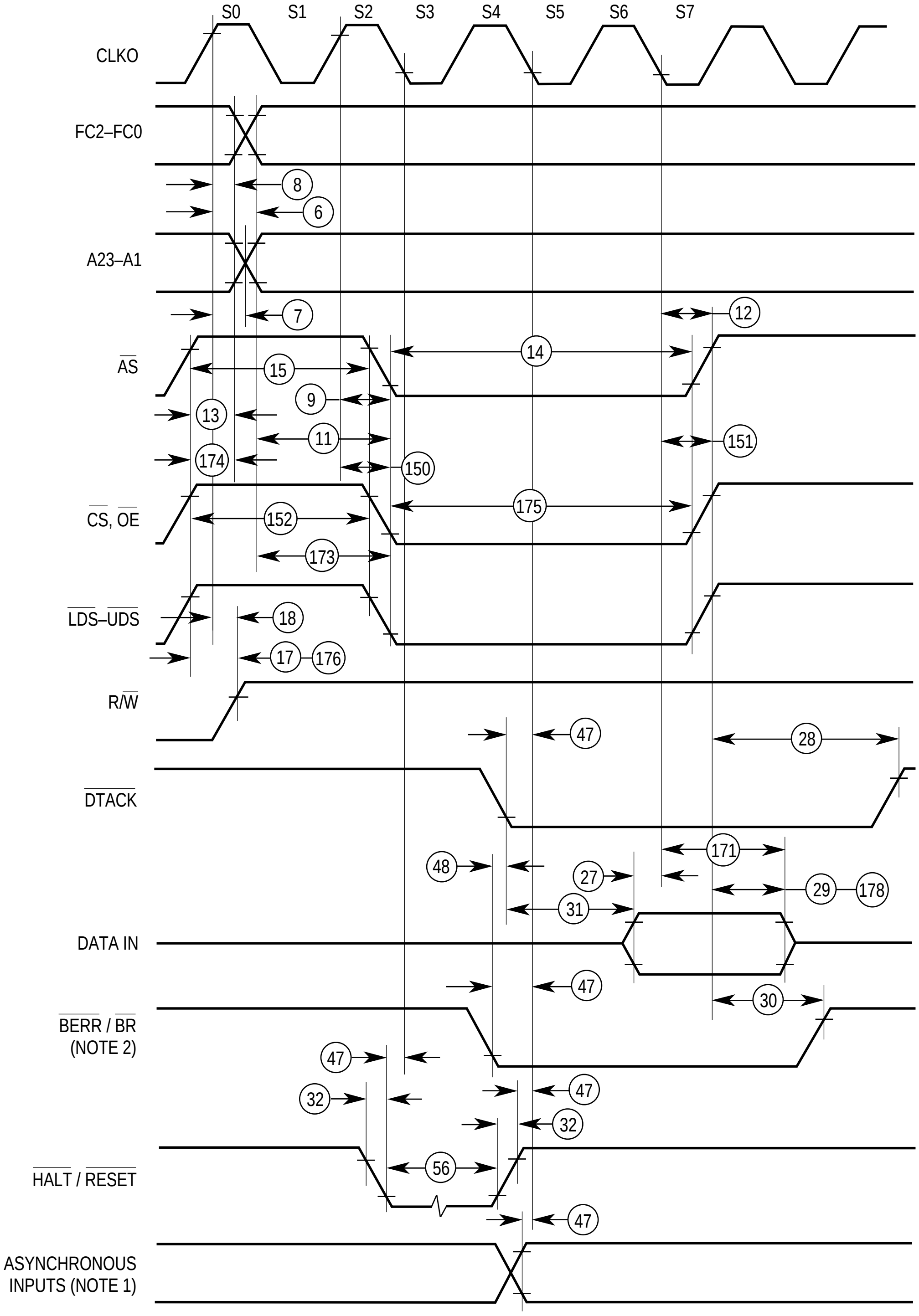
   Thus for spec 14 at 12.5 MHz:

   Sn = 5(80 ns/2) – 20 ns = 180 ns.

   These two formulas assume a 50% duty cycle. Otherwise, if N is odd, the previous values N(P/2) and N(Pa/2) must be reduced by X, where X is the difference between the nominal pulse width and the minimum pulse width of the EXTAL input clock for that duty cycle.
3. If #47 is satisfied for both $\overline{\text{DTACK}}$ and $\overline{\text{BERR}}$, #48 may be ignored. In the absence of $\overline{\text{DTACK}}$, $\overline{\text{BERR}}$ is a synchronous input using the asynchronous input setup time (#47).
4. For power-up, the IMP must be held in the reset state for a minimum 100 ms to allow stabilization of on-chip circuit. This time could be longer to allow the PLL to lock (see 7.6.1 IMP AC Electrical Specifications Control Timing. After the system is powered up, #56 refers to the minimum pulse width required to reset the processor.
5. If the asynchronous input setup (#47) requirement is satisfied for $\overline{\text{DTACK}}$, the $\overline{\text{DTACK}}$ asserted to data setup time (#31) requirement can be ignored. The data must only satisfy the data-in to clock low setup time (#27)

for the following clock cycle.

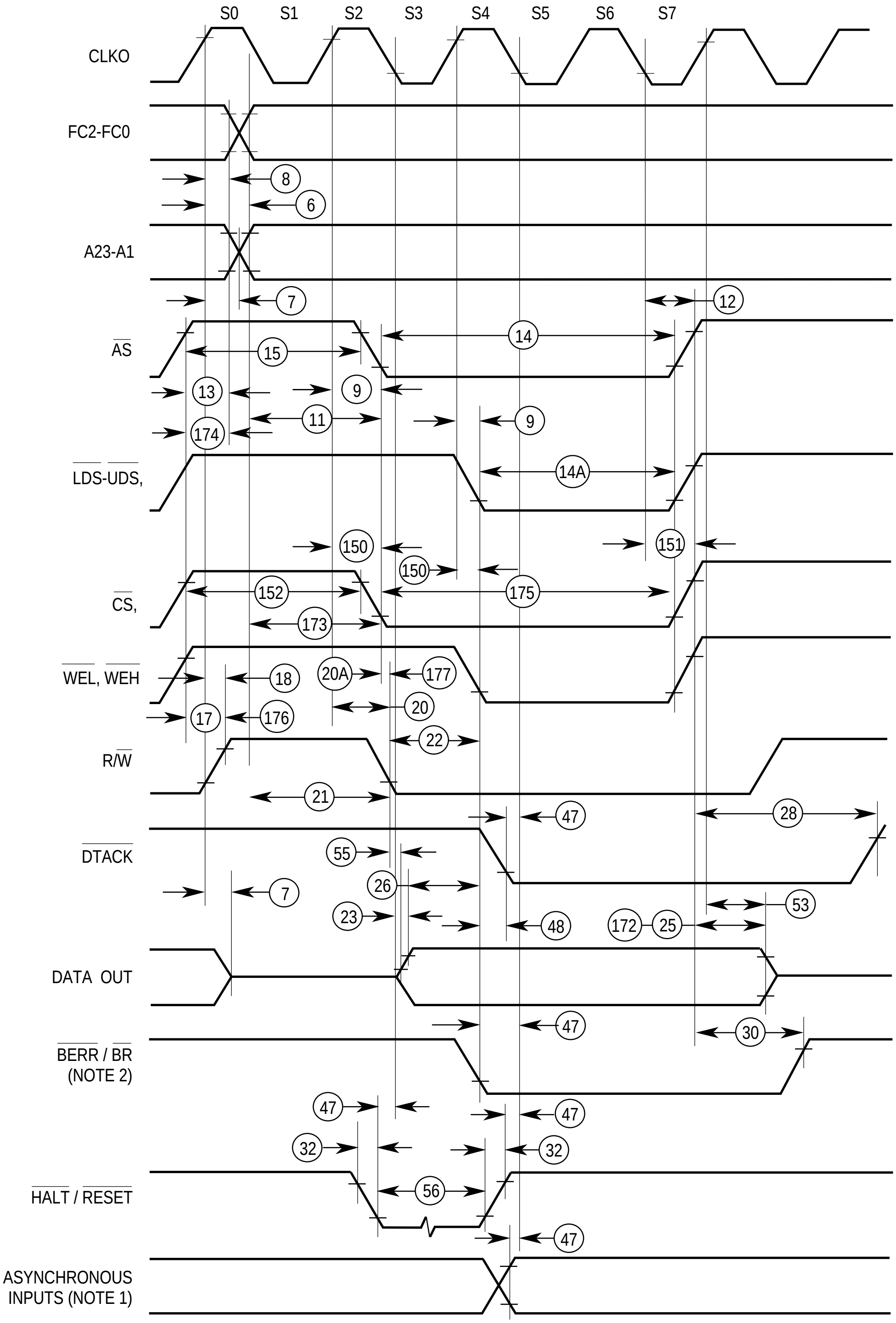
6. When $\overline{AS}$ and R/$\overline{W}$ are equally loaded (±20%), subtract 5 ns from the values given in these columns.
7. The MC68PM302 will negate $\overline{BG}$ and begin driving the bus if external arbitration logic negates $\overline{BR}$ before asserting $\overline{BGACK}$.
8. The minimum value must be met to guarantee proper operation. If the maximum value is exceeded, $\overline{BG}$ may be reasserted.



NOTES:

1. Setup time for the asynchronous inputs $\overline{\text{IPL2}}$–$\overline{\text{IPL0}}$ guarantees their recognition at the next falling edge of the clock.
2. $\overline{\text{BR}}$ need fall at this time only to insure being recognized at the end of the bus cycle.
3. Timing measurements are referenced to and from a low voltage of 0.8 volt and a high voltage of 2.0 volts, unless otherwise noted. The voltage swing through this range should start outside and pass through the range such that the rise or fall is linear between 0.8 volts and 2.0 volts.

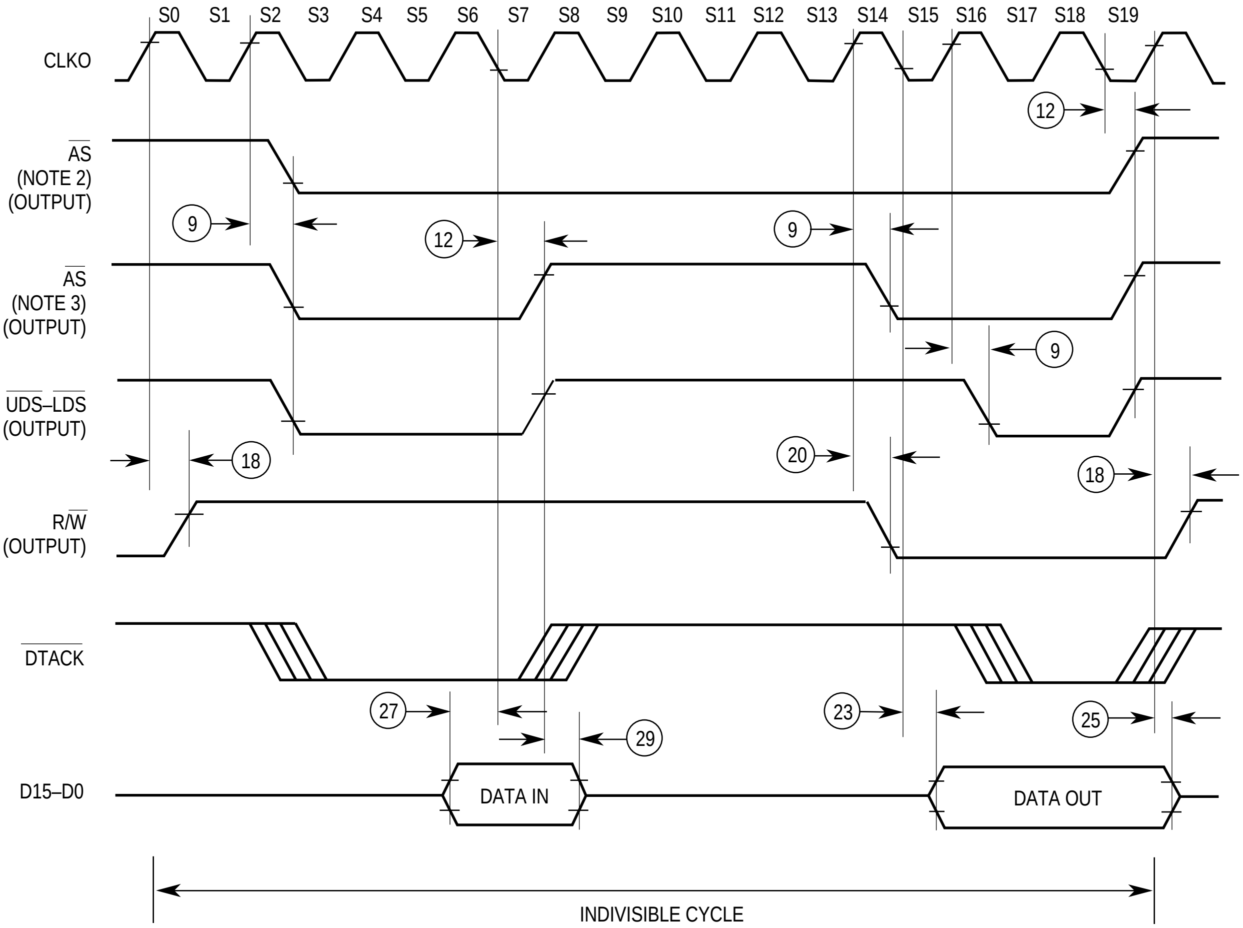
Figure 7-2. Read Cycle Timing Diagram



NOTES:

1. Timing measurements are referenced to and from a low voltage of 0.8 volt and a high voltage of 2.0 volts, unless otherwise noted. The voltage swing through this range should start outside and pass through the range such that the rise or fall is linear between between 0.8 volt and 2.0 volts.
2. Because of loading variations, R/W may be valid after AS even though both are initiated by the rising edge of S2 (specification #20A)
3. Each wait state is a full clock cycle inserted between S4 and S5.

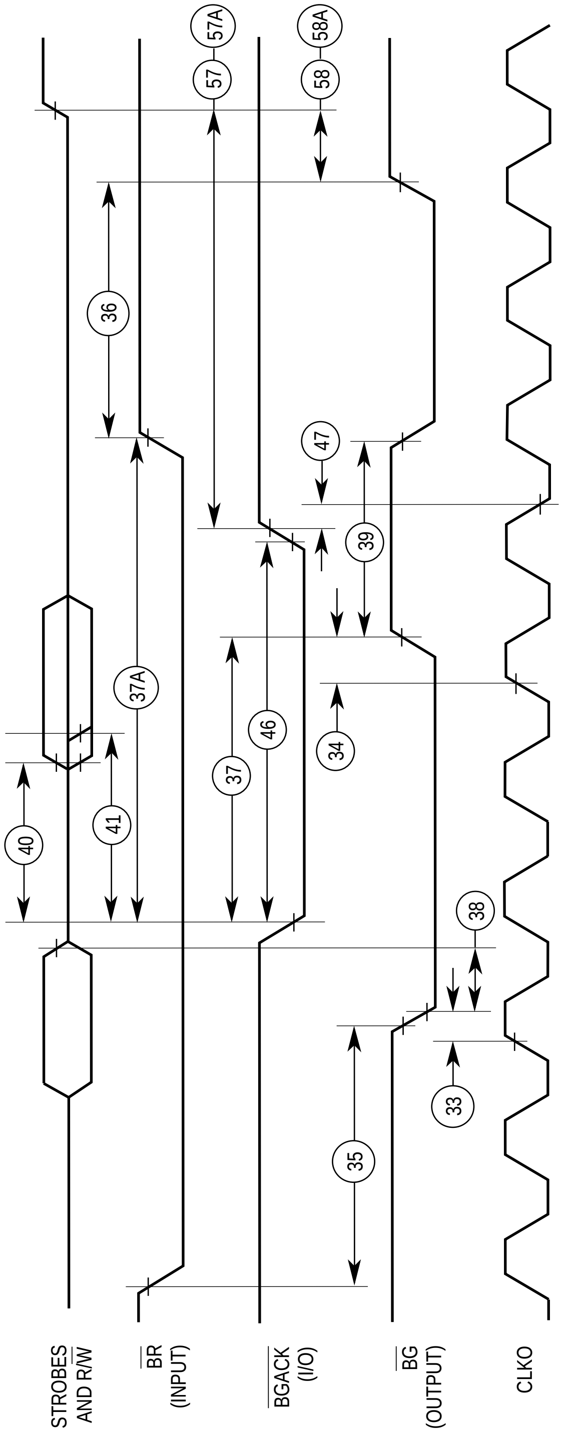
Figure 7-3. Write Cycle Timing Diagram



NOTES:

1. For other timings than indivisible cycles, see Figures 7-2 and 7-3.
2. RMCST = 0 in the SCR.
3. RMCST = 1 in the SCR.
4. Wait states may be inserted between S4 and S5 during the write cycle and between S16 and S17 during the read cycle.
5. Read-modify-write cycle is generated only by the TAS instruction.

Figure 7-4. Read-Modify-Write Cycle Timing Diagram



NOTE: Setup time to the clock (#47) for the asynchronous inputs $\overline{\text{BERR}}$, $\overline{\text{BGACK}}$, $\overline{\text{BR}}$, $\overline{\text{DTACK}}$, AND $\overline{\text{IPL2–IPL0}}$ guarantees their recognition at the next falling edge of the clock.

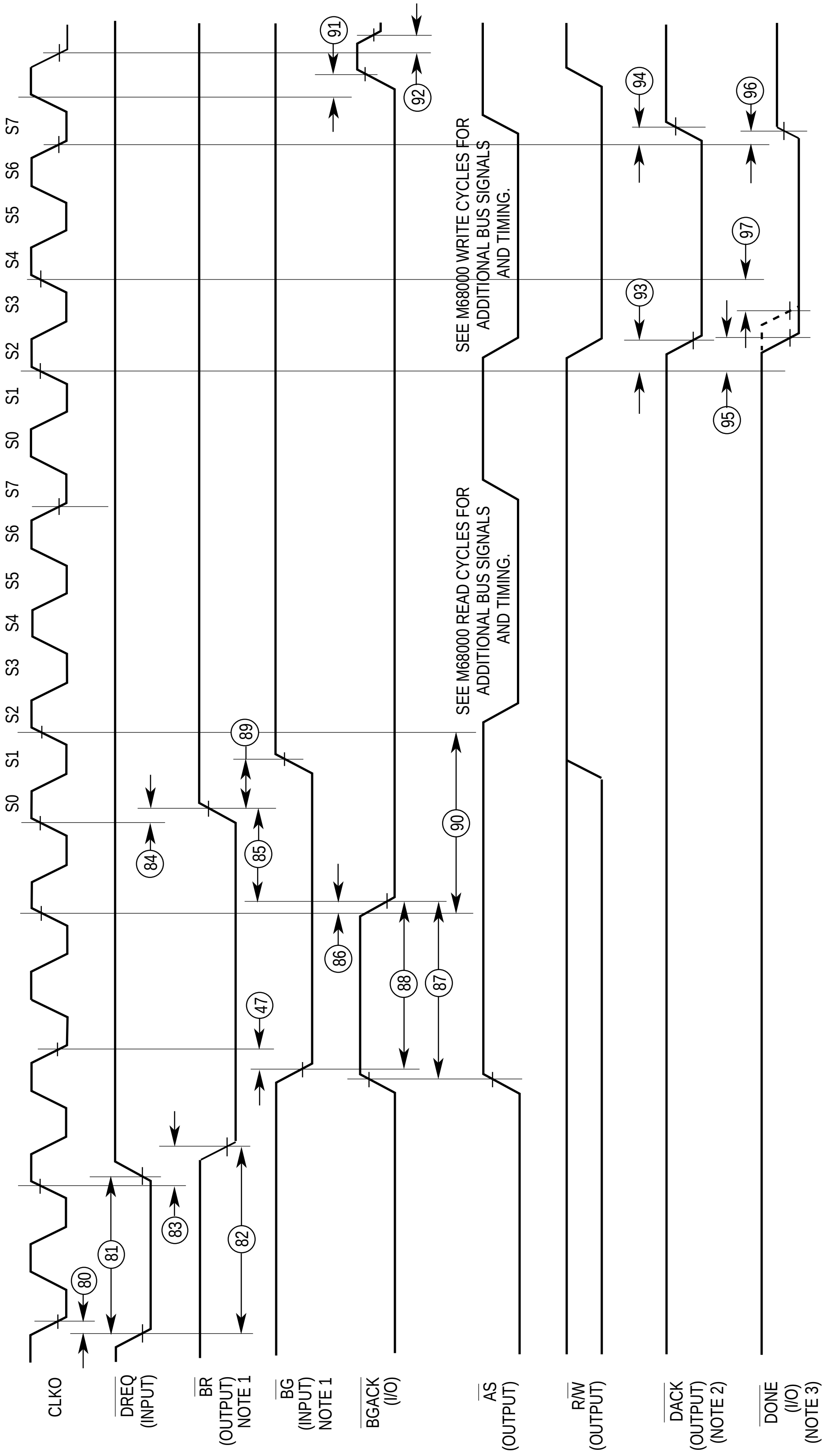
Figure 7-5. Bus Arbitration Timing Diagram

7.6.5 IMP AC Electrical Specifications—DMA (see Figure 7-6 and Figure 7-7)

Num.	Characteristic	Symbol	16.67 MHz at 5.0 V.		20 MHz at 5.0 V		25 MHz at 5.0 V		Unit
			Min	Max	Min	Max	Min	Max	
80	$\overline{\text{REQ}}$ Asynchronous Setup Time (see Note 1)	t_{REQASI}	15	—	15	—	10	—	ns
81	$\overline{\text{REQ}}$ Width Low (see Note 2)	t_{REQL}	2	—	2	—	2	—	clks
82	$\overline{\text{REQ}}$ Low to $\overline{\text{BR}}$ Low (see Notes 3 and 4)	$t_{REQLBRL}$	—	2	—	2	—	2	clks
83	Clock High to $\overline{\text{BR}}$ Low (see Notes 3 and 4)	t_{CHBRL}	—	30	—	25	—	20	ns
84	Clock High to $\overline{\text{BR}}$ High Impedance (see Notes 3 and 4)	t_{CHBRZ}	—	30	—	25	—	20	ns
85	$\overline{\text{BGACK}}$ Low to $\overline{\text{BR}}$ High Impedance (see Notes 3 and 4)	t_{BKLBRZ}	30	—	25	—	20	—	ns
86	Clock High to $\overline{\text{BGACK}}$ Low	t_{CHBKL}	—	30	—	25	—	20	ns
87	$\overline{\text{AS}}$ and $\overline{\text{BGACK}}$ High (the Latest One) to $\overline{\text{BGACK}}$ Low (when $\overline{\text{BG}}$ Is Asserted)	t_{ABHBKL}	1.5	2.5 +30	1.5	2.5 +25	1.5	2.5 +20	clks ns
88	$\overline{\text{BG}}$ Low to $\overline{\text{BGACK}}$ Low (No Other Bus Master) (see Notes 3 and 4)	t_{BGLBKL}	1.5	2.5 +30	1.5	2.5 +25	1.5	2.5 +20	clks ns
89	$\overline{\text{BR}}$ High Impedance to $\overline{\text{BG}}$ High (see Notes 3 and 4)	t_{BRHBGH}	0	—	0	—	0	—	ns
90	Clock on which $\overline{\text{BGACK}}$ Low to Clock on which $\overline{\text{AS}}$ Low	$t_{CLBKLAL}$	2	2	2	2	2	2	clks
91	Clock High to $\overline{\text{BGACK}}$ High	t_{CHBKH}	—	30	—	25	—	20	ns
92	Clock Low to $\overline{\text{BGACK}}$ High Impedance	t_{CLBKZ}	—	15	—	15	—	10	ns
93	Clock High to $\overline{\text{DACK}}$ Low	t_{CHACKL}	—	30	—	25	—	20	ns
94	Clock Low to $\overline{\text{DACK}}$ High	t_{CLACKH}	—	30	—	25	—	20	ns
95	Clock High to $\overline{\text{DONE}}$ Low (Output)	t_{CHDNL}	—	30	—	25	—	20	ns
96	Clock Low to $\overline{\text{DONE}}$ High Impedance	t_{CLDNZ}	—	30	—	25	—	20	ns
97	$\overline{\text{DONE}}$ Input Low to Clock High (Asynchronous Setup)	t_{DNLTCH}	15	—	15	—	10	—	ns

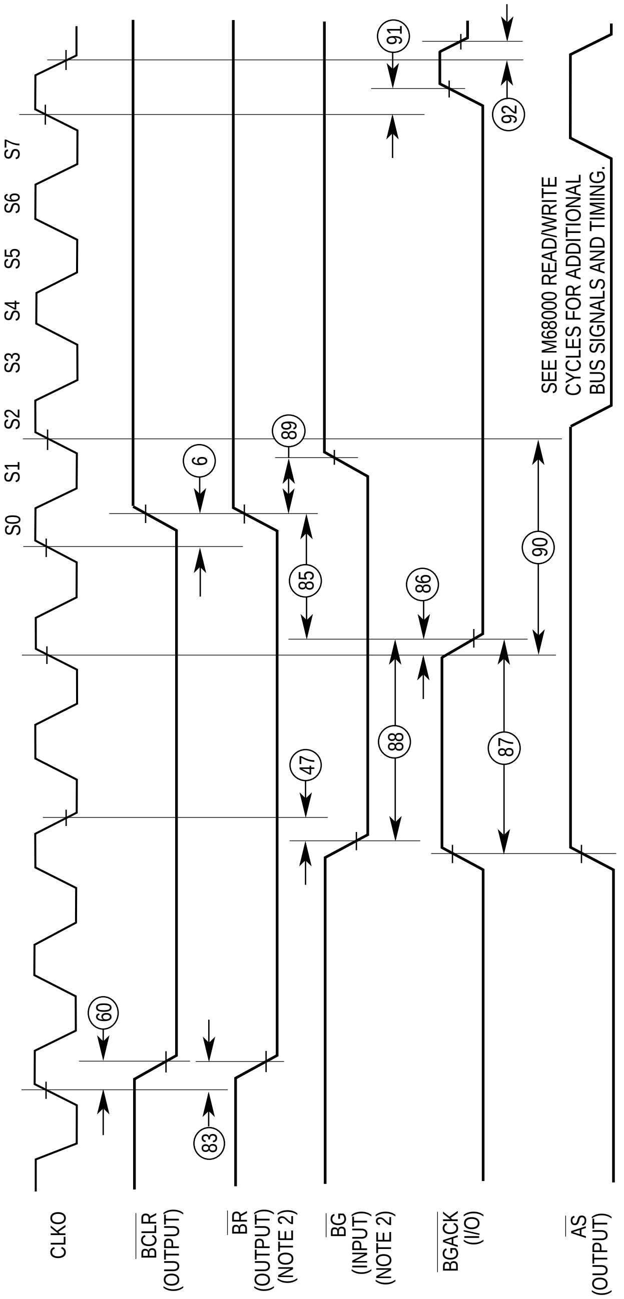
NOTES:

1. $\overline{\text{DREQ}}$ is sampled on the falling edge of $\overline{\text{CLK}}$ in cycle steal and burst modes.
2. If #80 is satisfied for $\overline{\text{DREQ}}$, #81 may be ignored.
3. $\overline{\text{BR}}$ will not be asserted while $\overline{\text{AS}}$, $\overline{\text{HALT}}$, or $\overline{\text{BERR}}$ is asserted.
4. Specifications are for DISABLE CPU mode only.
5. $\overline{\text{DREQ}}$, $\overline{\text{DACK}}$, and $\overline{\text{DONE}}$ do not apply to the SDMA channels.
6. DMA and SDMA read and write cycle timing is the same as that for the M68000 core.



NOTES:
1. BR and BG shown above are only active in disable CPU mode; otherwise, they do not apply to the diagram.
2. Assumes the ECO bit in the CMR = 1.
3. For the case when DONE is an input, assumes ECO bit in the CMR = 1.

Figure 7-6. DMA Timing Diagram (IDMA)


S0
S1
S2
S3
S4
S5
S6
S7
CLKO
$\overline{\text{BCLR}}$ (OUTPUT)
$\overline{\text{BR}}$ (OUTPUT) (NOTE 2)
$\overline{\text{BG}}$ (INPUT) (NOTE 2)
$\overline{\text{BGACK}}$ (I/O)
$\overline{\text{AS}}$ (OUTPUT)
60
83
47
6
89
85
86
88
87
90
91
92
SEE M68000 READ/WRITE CYCLES FOR ADDITIONAL BUS SIGNALS AND TIMING.

NOTES:
1. DRAM refresh controller timing is identical to SDMA timing.
2. $\overline{\text{BR}}$ and $\overline{\text{BG}}$ shown above are only active in disable CPU mode; otherwise they do not apply to the diagram.

Figure 7-7. DMA Timing Diagram (SDMA)

7.6.6 IMP AC Electrical Specifications—External Master Internal Asynchronous Read/Write Cycles

(see Figure 7-8 and Figure 7-9)

Num.	Characteristic	Symbol	16.67 MHz at 5.0 V		20 MHz at 5.0 V		25 MHz at 5.0 V		Unit
			Min	Max	Min	Max	Min	Max	
100	R/$\overline{\text{W}}$ Valid to $\overline{\text{DS}}$ Low	t_{RWVDSL}	0	—	0	—	0	—	ns
101	$\overline{\text{DS}}$ Low to Data-In Valid	t_{DSLDIV}	—	30	—	25	—	20	ns
102	$\overline{\text{DTACK}}$ Low to Data-In Hold Time	t_{DKLDH}	0	—	0	—	0	—	ns
103	$\overline{\text{AS}}$ Valid to $\overline{\text{DS}}$ Low	t_{ASVDSL}	0	—	0	—	0	—	ns
104	$\overline{\text{DTACK}}$ Low to $\overline{\text{AS}}$, $\overline{\text{DS}}$ High	t_{DKLDSH}	0	—	0	—	0	—	ns
105	$\overline{\text{DS}}$ High to $\overline{\text{DTACK}}$ High	t_{DSHDKH}	—	45	—	40	—	30	ns
106	$\overline{\text{DS}}$ Inactive to $\overline{\text{AS}}$ Inactive	t_{DSIASI}	0	—	0	—	0	—	ns
107	$\overline{\text{DS}}$ High to R/$\overline{\text{W}}$ High	t_{DSHRWH}	0	—	0	—	0	—	ns
108	$\overline{\text{DS}}$ High to Data High Impedance	t_{DSHDZ}	—	45	—	40	—	30	ns
108A	$\overline{\text{DS}}$ High to Data-Out Hold Time (see Note)	t_{DSHDH}	0	—	0	—	0	—	ns
109A	Data Out Valid to $\overline{\text{DTACK}}$ Low	t_{DOVDKL}	15	—	15	—	10	—	ns

NOTE: If $\overline{\text{AS}}$ is negated before $\overline{\text{DS}}$, the data bus could be three-stated (spec 126) before $\overline{\text{DS}}$ is negated.

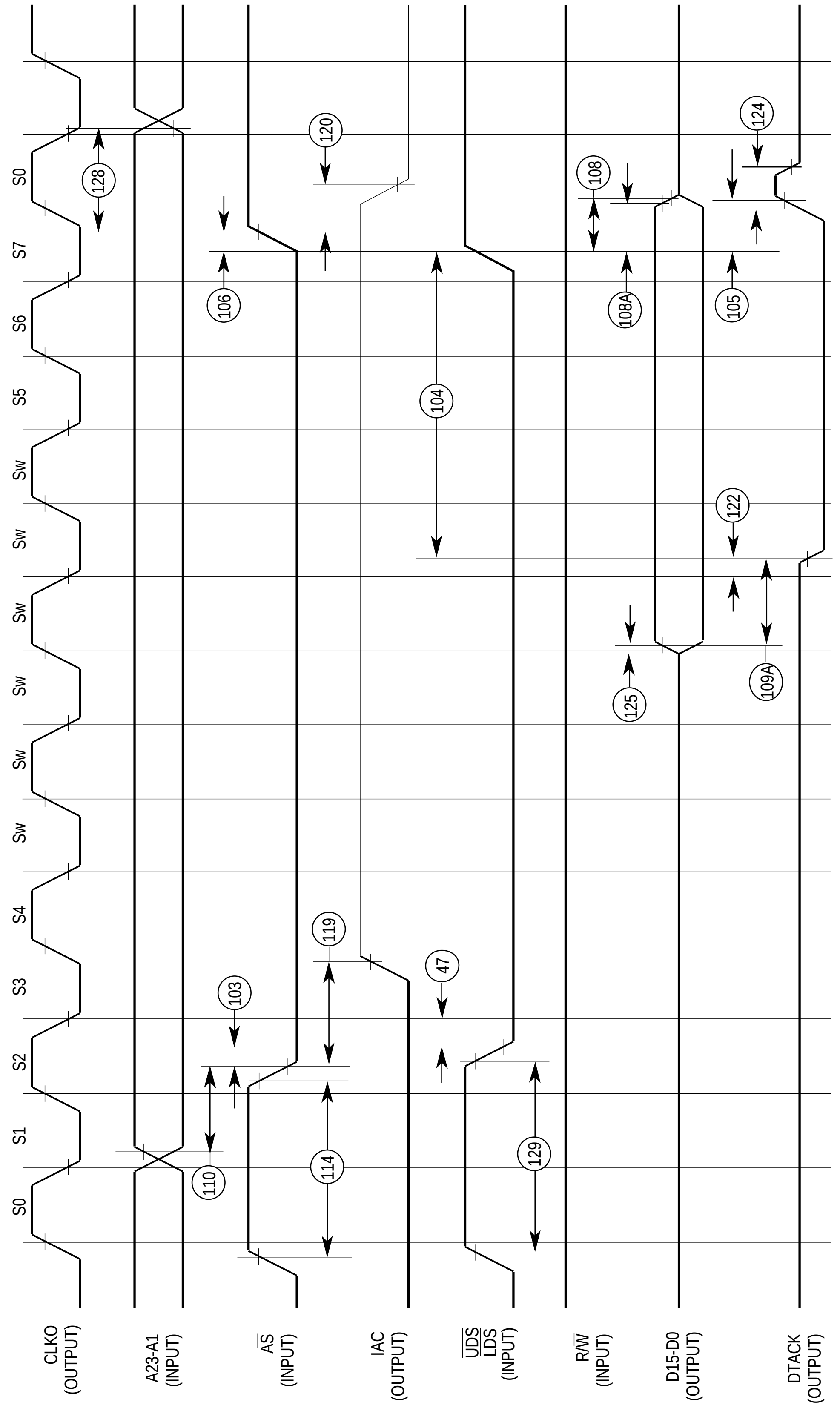


Figure 7-8. External Master Internal Asynchronous Read Cycle Timing Diagram

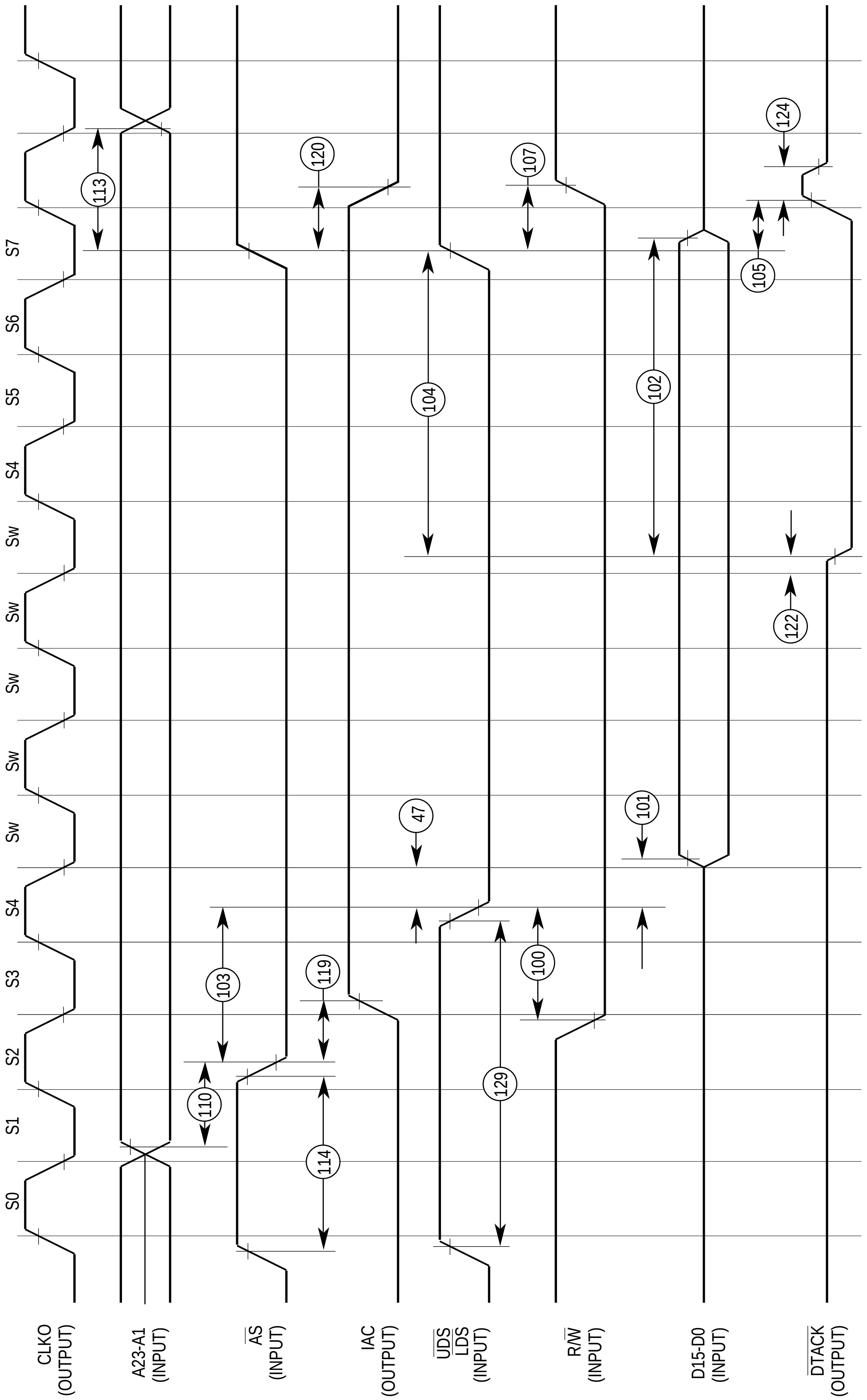


Figure 7-9. External Master Internal Asynchronous Write Cycle Timing Diagram

7.6.7 IMP AC Electrical Specifications—External Master Internal Synchronous Read/Write Cycles

(see Figure 7-10, Figure 7-11, and Figure 7-12)

Num.	Characteristic	Symbol	16.67 MHz at 5.0 V		20 MHz at 5.0 V		25 MHz at 5.0 V		Unit
			Min	Max	Min	Max	Min	Max	
110	Address Valid to $\overline{\text{AS}}$ Low	t_{AVASL}	15	—	12	—	10	—	ns
111	$\overline{\text{AS}}$ Low to Clock High	t_{ASLCH}	30	—	25	—	20	—	ns
112	Clock Low to $\overline{\text{AS}}$ High	t_{CLASH}	—	45	—	40	—	30	ns
113	$\overline{\text{AS}}$ High to Address Hold Time on Write	t_{ASHAH}	0	—	0	—	0	—	ns
114	$\overline{\text{AS}}$ Inactive Time	t_{ASH}	1	—	1	—	1	—	clk
115	$\overline{\text{UDS}}$/$\overline{\text{LDS}}$ Low to Clock High (see Note 2)	t_{SLCH}	40	—	33	—	27	—	ns
116	Clock Low to $\overline{\text{UDS}}$/$\overline{\text{LDS}}$ High	t_{CLSH}	—	45	—	40	—	30	ns
117	R/$\overline{\text{W}}$ Valid to Clock High (see Note 2)	t_{RWVCH}	30	—	25	—	20	—	ns
118	Clock High to R/$\overline{\text{W}}$ High	t_{CHRWH}	—	45	—	40	—	30	ns
119	$\overline{\text{AS}}$ Low to IAC High	t_{ASLIAH}	—	40	—	35	—	27	ns
120	$\overline{\text{AS}}$ High to IAC Low	t_{ASHIAL}	—	40	—	35	—	27	ns
121	$\overline{\text{AS}}$ Low to $\overline{\text{DTACK}}$ Low (0 Wait State)	t_{ASLDTL}	—	45	—	40	—	30	ns
122	Clock Low to $\overline{\text{DTACK}}$ Low (1 Wait State)	t_{CLDTL}	—	30	—	25	—	20	ns
123	$\overline{\text{AS}}$ High to $\overline{\text{DTACK}}$ High	t_{ASHDTH}	—	45	—	40	—	30	ns
124	$\overline{\text{DTACK}}$ High to $\overline{\text{DTACK}}$ High Impedance	t_{DTHDTZ}	—	15	—	15	—	10	ns
125	Clock High to Data-Out Valid	t_{CHDOV}	—	30	—	25	—	20	ns
126	$\overline{\text{AS}}$ High to Data High Impedance	t_{ASHDZ}	—	45	—	40	—	30	ns
127	$\overline{\text{AS}}$ High to Data-Out Hold Time	t_{ASHDOI}	0	—	0	—	0	—	ns
128	$\overline{\text{AS}}$ High to Address Hold Time on Read	t_{ASHAI}	0	—	0	—	0	—	ns
129	$\overline{\text{UDS}}$/$\overline{\text{LDS}}$ Inactive Time	t_{SH}	1	—	1	—	1	—	clk
130	Data-In Valid to Clock Low	t_{CLDIV}	30	—	25	—	20	—	ns
131	Clock Low to Data-In Hold Time	t_{CLDIH}	15	—	12	—	10	—	ns

NOTES:

1. Synchronous specifications above are valid only when SAM = 1 in the SCR.
2. It is required that this signal not be asserted prior to the previous rising CLKO edge (i.e., in the previous clock cycle). It must be recognized by the IMP no sooner than the rising CLKO edge shown in the diagram.

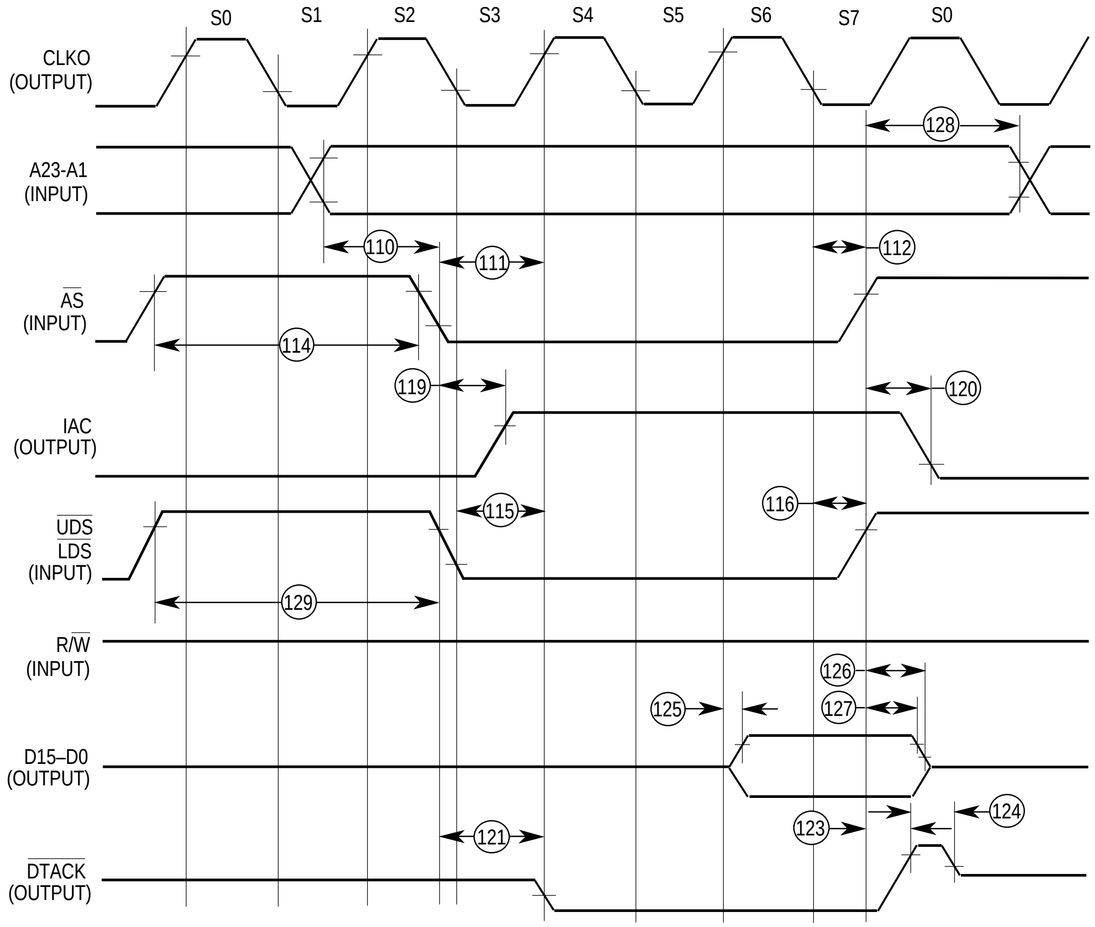


Figure 7-10. External Master Internal Synchronous Read Cycle Timing Diagram

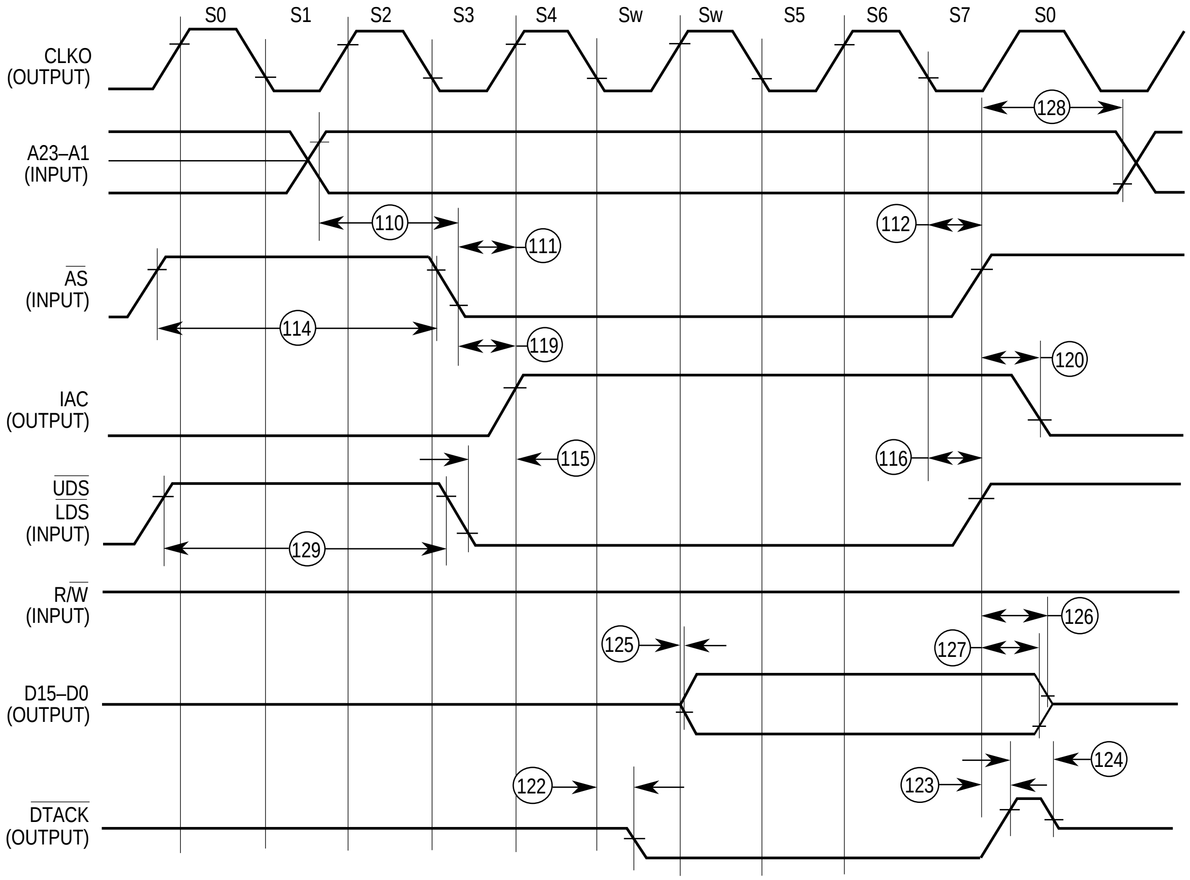


Figure 7-11. External Master Internal Synchronous Read Cycle Timing Diagram (One Wait State)

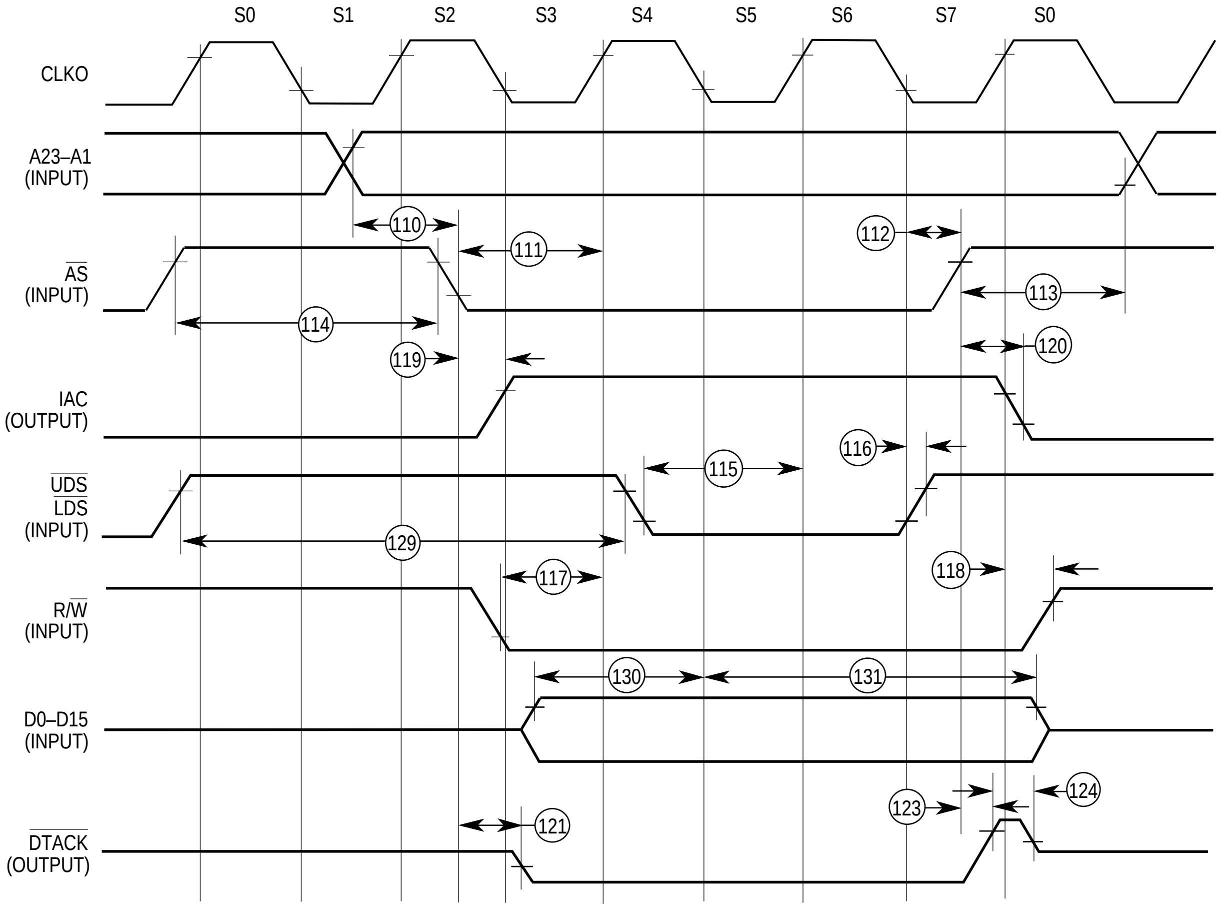


Figure 7-12. External Master Internal Synchronous Write Cycle Timing Diagram

7.6.8 IMP AC Electrical Specifications—Internal Master Internal Read/Write Cycles (see Figure 7-13)

Num.	Characteristic	Symbol	16.67 MHz at 5.0 V		20 MHz at 5.0 V		25 MHz at 5.0 V		Unit
			Min	Max	Min	Max	Min	Max	
140	Clock High to IAC High	t_{CHIAH}	—	40	—	35	—	27	ns
141	Clock Low to IAC Low	t_{CLIAL}	—	40	—	35	—	27	ns
142	Clock High to $\overline{\text{DTACK}}$ Low	t_{CHDTL}	—	45	—	40	—	30	ns
143	Clock Low to $\overline{\text{DTACK}}$ High	t_{CLDTH}	—	40	—	35	—	27	ns
144	Clock High to Data-Out Valid	t_{CHDOV}	—	30	—	25	—	20	ns
145	$\overline{\text{AS}}$ High to Data-Out Hold Time	t_{ASHDOH}	0	—	0	—	0	—	ns

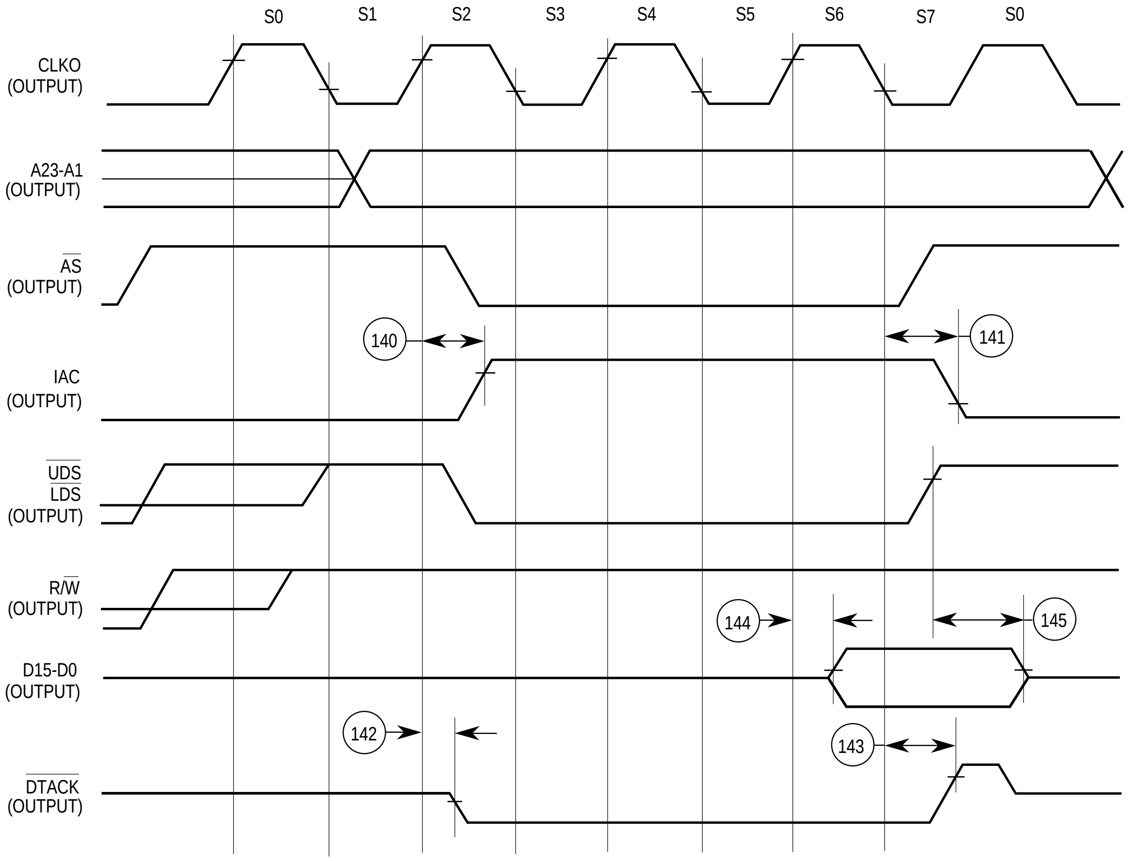


Figure 7-13. Internal Master Internal Read Cycle Timing Diagram

7.6.9 IMP AC Electrical Specifications—Chip-Select Timing Internal Master (see Figure 7-14)

NOTE:

1. This specification is valid only when the ADCE or WPVE bits in the SCR are set.
2. For loading capacitance less than or equal to 50 pF, subtract 4 ns from the maximum value given.
3. Since $\overline{\text{AS}}$ and $\overline{\text{CS}}$ are asserted/negated on the same CLKO edges, no $\overline{\text{AS}}$ to $\overline{\text{CS}}$ relative timings can be

Num.	Characteristic	Symbol	16.67 MHz at 5.0 V		20 MHz at 5.0 V		25 MHz at 5.0 V		Unit
			Min	Max	Min	Max	Min	Max	
150	Clock High to $\overline{CS}$, $\overline{IACK}$, $\overline{OE}$, $\overline{WEL}$, $\overline{WEH}$ Low (see Note 2)	$t_{CHCSIAKL}$	0	40	0	35	0	27	ns
151	Clock Low to $\overline{CS}$, $\overline{IACK}$, $\overline{OE}$, $\overline{WEL}$, $\overline{WEH}$ High (see Note 2)	$t_{CLCSIAKH}$	0	40	0	35	0	27	ns
152	$\overline{CS}$ Width Negated	t_{CSH}	60	—	50	—	40	—	ns
153	Clock High to $\overline{DTACK}$ Low (0 Wait State)	t_{CHDTKL}	—	45	—	40	—	30	ns
154	Clock Low to $\overline{DTACK}$ Low (1–6 Wait States)	t_{CLDTKL}	—	30	—	25	—	20	ns
155	Clock Low to $\overline{DTACK}$ High	t_{CLDTKH}	—	40	—	35	—	27	ns
156	Clock High to $\overline{BERR}$ Low (see Note 1)	t_{CHBERL}	—	40	—	35	—	27	ns
157	Clock Low to $\overline{BERR}$ High Impedance (see Note 1)	t_{CLBERH}	—	40	—	35	—	27	ns
158	$\overline{DTACK}$ High to $\overline{DTACK}$ High Impedance	$t_{DTKHDTKZ}$	—	15	—	15	—	10	ns
171	Input Data Hold Time from S6 Low	t_{IDHCL}	5	—	5	—	5	—	ns
172	$\overline{CS}$ Negated to Data-Out Invalid (Write)	t_{CSNDOI}	10	—	10	—	7	—	ns
173	Address, FC Valid to $\overline{CS}$ Asserted	t_{AFVCSA}	15	—	15	—	15	—	ns
174	$\overline{CS}$ Negated to Address, FC Invalid	t_{CSNAFI}	15	—	15	—	12	—	ns
175	$\overline{CS}$ Low Time (0 Wait States)	t_{CSLT}	120	—	100	—	80	—	ns
176	$\overline{CS}$ Negated to R/$\overline{W}$ Invalid	t_{CSNRWI}	10	—	10	—	7	—	ns
177	$\overline{CS}$ Asserted to R/$\overline{W}$ Low (Write)	t_{CSARWL}	—	10	—	10	—	8	ns
178	$\overline{CS}$ Negated to Data-In Invalid (Hold Time on Read)	t_{CSNDII}	0	—	0	—	0	—	ns

specified. However, $\overline{CS}$ timings are given relative to a number of other signals, in the same manner as $\overline{AS}$. See Figure 7-2 and Figure 7-3 for diagrams.

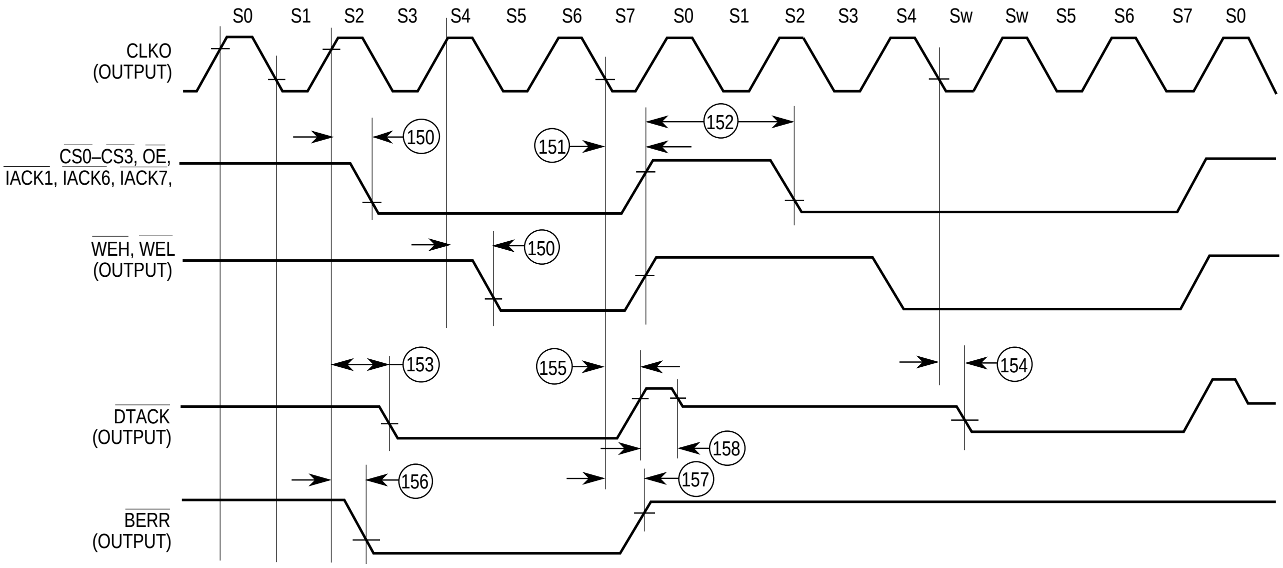


Figure 7-14. Internal Master Chip-Select Timing Diagram

7.6.10 IMP AC Electrical Specifications—Chip-Select Timing External Master (see Figure 7-15)

Num.	Characteristic	Symbol	16.67 MHz at 5.0 V		20 MHz at 5.0 V		25 MHz at 5.0 V		Unit
			Min	Max	Min	Max	Min	Max	
154	Clock Low to $\overline{\text{DTACK}}$ Low (1–6 Wait States)	t_{CLDTKL}	—	30	—	25	—	20	ns
160	$\overline{\text{AS}}$ Low to $\overline{\text{CS}}$ Low	t_{ASLCSL}	—	30	—	25	—	20	ns
161	$\overline{\text{AS}}$ High to $\overline{\text{CS}}$ High	t_{ASHCSH}	—	30	—	25	—	20	ns
162	Address Valid to $\overline{\text{AS}}$ Low	t_{AVASL}	15	—	12	—	10	—	ns
163	R/$\overline{\text{W}}$ Valid to $\overline{\text{AS}}$ Low (see Note 1)	t_{RWVASL}	15	—	12	—	10	—	ns
164	$\overline{\text{AS}}$ Negated to Address Hold Time	t_{ASHAI}	0	—	0	—	0	—	ns
165	$\overline{\text{AS}}$ Low to $\overline{\text{DTACK}}$ Low (0 Wait State)	$t_{ASLDTKL}$	—	45	—	40	—	30	ns
167	$\overline{\text{AS}}$ High to $\overline{\text{DTACK}}$ High	$t_{ASHDTKH}$	—	30	—	25	—	20	ns
168	$\overline{\text{AS}}$ Low to $\overline{\text{BERR}}$ Low (see Note 2)	$t_{ASLBERL}$	—	30	—	25	—	20	ns
169	$\overline{\text{AS}}$ High to $\overline{\text{BERR}}$ High Impedance (see Note 2)	$t_{ASHBERH}$	—	30	—	25	—	20	ns

NOTES:
1. The minimum value must be met to guarantee write protection operation.
2. This specification is not applicable to the 68PM302. BERR_ is only an input pin. The BERR_ signal is internally asserted to the CPU if ADCE or WPVE bits in the SCR are set.

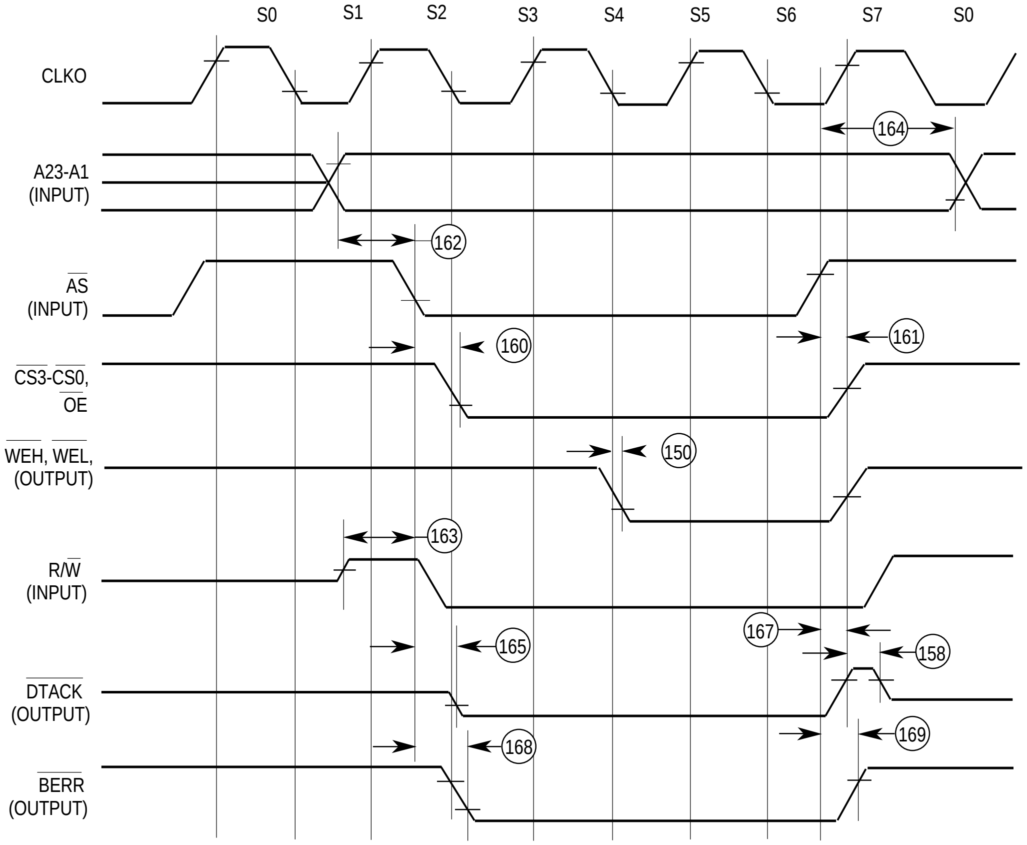


Figure 7-15. External Master Chip-Select Timing Diagram

7.6.11 IMP AC Electrical Specifications—Parallel I/O (see Figure 7-16)

Num.	Characteristic	Symbol	16.67 MHz at 5.0 V		20 MHz at 5.0 V		25 MHz at 5.0 V		Unit
			Min	Max	Min	Max	Min	Max	
180	Input Data Setup Time (to Clock Low)	t_{DSU}	20	—	20	—	14	—	ns
181	Input Data Hold Time (from Clock Low)	t_{DH}	10	—	10	—	19	—	ns
182	Clock High to Data-Out Valid (CPU Writes Data, Control, or Direction)	t_{CHDOV}	—	35	—	30	—	24	ns

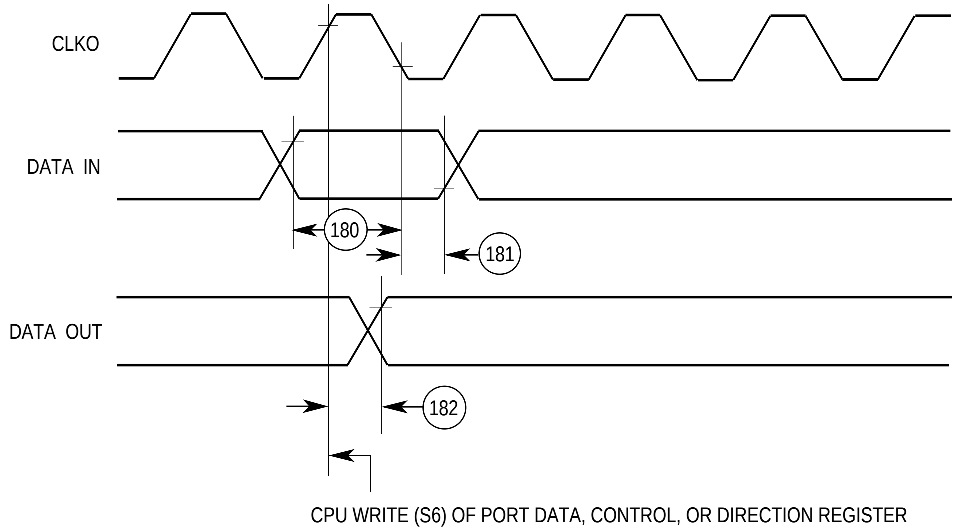


Figure 7-16. Parallel I/O Data-In/Data-Out Timing Diagram

7.6.12 IMP AC Electrical Specifications—Interrupts (see Figure 7-17)

Num.	Characteristic	Symbol	16.67 MHz at 5.0 V		20 MHz at 5.0 V		25 MHz at 5.0 V		Unit
			Min	Max	Min	Max	Min	Max	
190	Interrupt Pulse Width Low $\overline{IRQ}$ (Edge Triggered Mode)	t_{PW}	50	—	42	—	34	—	ns
191	Minimum Time Between Active Edges	t_{AEMT}	3	—	3	—	3	—	clk

NOTE: Setup time for the asynchronous inputs $\overline{IPL2}$–$\overline{IPL0}$ and $\overline{AVEC}$ guarantees their recognition at the next falling edge of the clock.

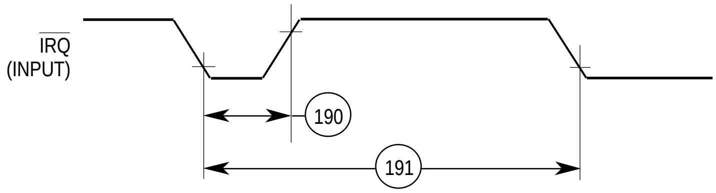


Figure 7-17. Interrupts Timing Diagram

7.6.13 IMP AC Electrical Specifications—Timers (see Figure 7-18)

Num.	Characteristic	Symbol	16.67 MHz at 5.0 V		20 MHz at 5.0 V		25 MHz at 5.0 V		Unit
			Min	Max	Min	Max	Min	Max	
200	Timer Input Capture Pulse Width	t_{TPW}	50	—	42	—	34	—	ns
201	TIN Clock Low Pulse Width	t_{TICLT}	50	—	42	—	34	—	ns
202	TIN Clock High Pulse Width and Input Capture High Pulse Width	t_{TICHT}	2	—	2	—	2	—	clk
203	TIN Clock Cycle Time	t_{cyc}	3	—	3	—	3	—	clk
204	Clock High to TOUT Valid	t_{CHTOV}	—	35	—	30	—	24	ns
205	$\overline{\text{FRZ}}$ Input Setup Time (to Clock High) (see Note 1)	t_{FRZSU}	20	—	20	—	14	—	ns
206	$\overline{\text{FRZ}}$ Input Hold Time (from Clock High)	t_{FRZHT}	10	—	10	—	7	—	ns

NOTES:
1. $\overline{\text{FRZ}}$ should be negated during total system reset.
2. The TIN specs above do not apply to the use of TIN1 as a baud rate generator input clock. In such a case, specifications 1–3 may be used.

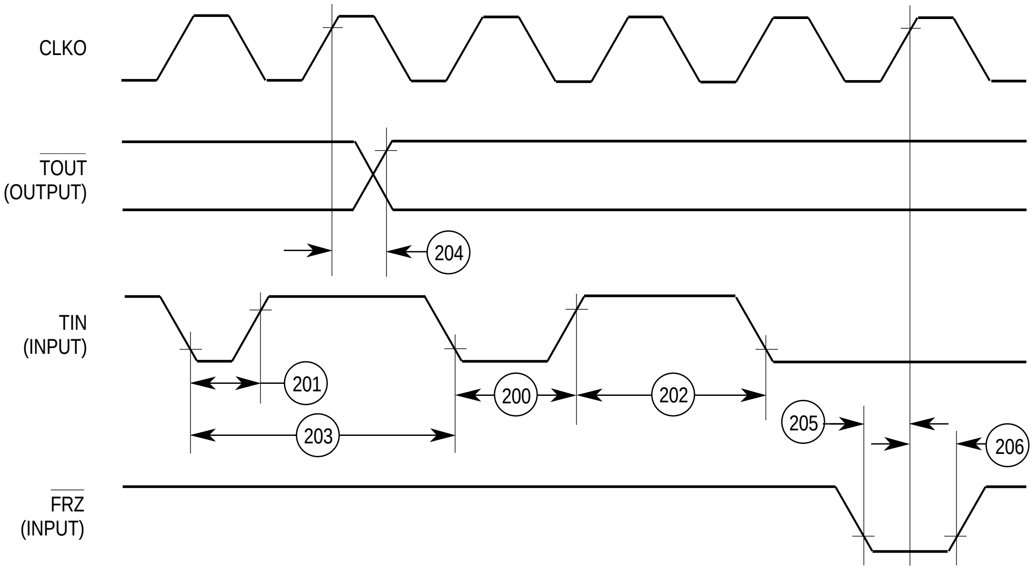


Figure 7-18. Timers Timing Diagram

7.6.14 IMP AC Electrical Specifications—Serial Communications Port (see Figure 7-19).

Num.	Characteristic	16.67 MHz at 5.0 V		20 MHz at 5.0 V		20 MHz at 5.0 V		Unit
		Min	Max	Min	Max	Min	Max	
250	SPCLK Clock Output Period	4	64	4	64	4	64	clks
251	SPCLK Clock Output Rise/Fall Time	0	15	0	10	0	8	ns
252	Delay from SPCLK to Transmit (see Note 1)	0	40	0	30	0	24	ns
253	SCP Receive Setup Time (see Note 1)	40	—	30	—	24	—	ns
254	SCP Receive Hold Time (see Note 1)	10	—	8	—	7	—	ns

NOTES:
1. This also applies when SPCLK is inverted by CI in the SPMODE register.
2. The enable signals for the slaves may be implemented by the parallel I/O pins.

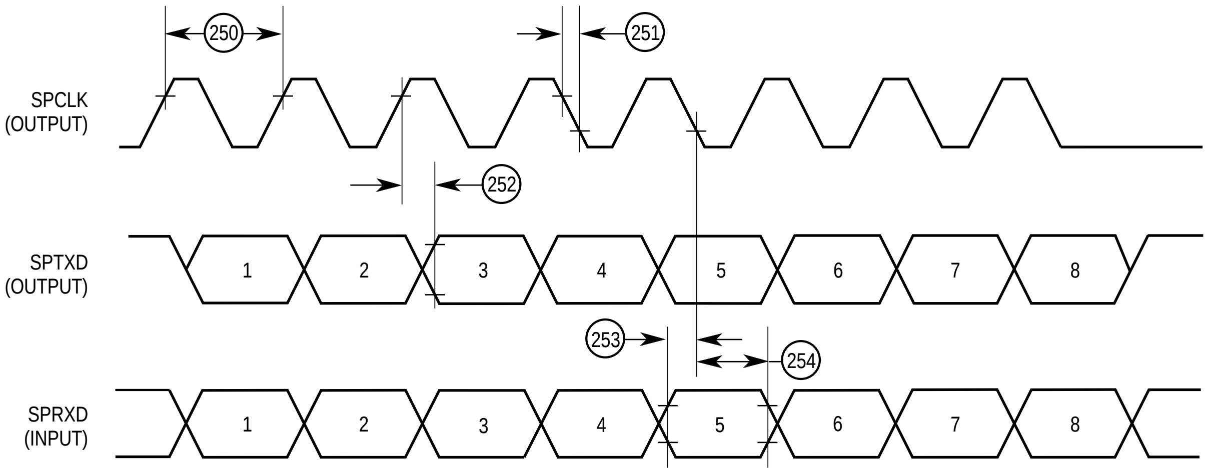


Figure 7-19. Serial Communication Port Timing Diagram

7.6.15 IMP AC Electrical Specifications—IDL Timing (All timing measurements, unless otherwise specified, are referenced to the L1CLK at 50% point of V_{DD}) (see Figure 7-20)

Num.	Characteristic	16.67 MHz at 5.0 V		20 MHz at 5.0 V		25 MHz at 5.0 V		Unit
		Min	Max	Min	Max	Min	Max	
260	L1CLK (IDL Clock) Frequency (see Note 1)	—	6.66	—	8	—	10	MHz
261	L1CLK Width Low	55	—	45	—	37	—	ns
262	L1CLK Width High (see Note 3)	P+10	—	P+10	—	P+10	—	ns
263	L1TXD, L1RQ, SDS1–SDS2 Rising/Falling Time	—	20	—	17	—	14	ns
264	L1SY1 (sync) Setup Time (to L1CLK Falling Edge)	30	—	25	—	20	—	ns
265	L1SY1 (sync) Hold Time (from L1CLK Falling Edge)	50	—	40	—	34	—	ns
266	L1SY1 (sync) Inactive Before 4th L1CLK	0	—	0	—	0	—	ns
267	L1TxD Active Delay (from L1CLK Rising Edge)	0	75	0	65	0	50	ns
268	L1TxD to High Impedance (from L1CLK Rising Edge) (see Note 2)	0	50	0	42	0	34	ns
269	L1RxD Setup Time (to L1CLK Falling Edge)	50	—	42	—	34	—	ns
270	L1RxD Hold Time (from L1CLK Falling Edge)	50	—	42	—	34	—	ns
271	Time Between Successive IDL syncs	20	—	20	—	20	—	L1CLK
272	L1RQ Valid before Falling Edge of L1SY1	1	—	1	—	1	—	L1CLK
273	L1GR Setup Time (to L1SY1 Falling Edge)	50	—	42	—	34	—	ns
274	L1GR Hold Time (from L1SY1 Falling Edge)	50	—	42	—	34	—	ns
275	SDS1–SDS2 Active Delay from L1CLK Rising Edge	10	75	10	65	7	50	ns
276	SDS1–SDS2 Inactive Delay from L1CLK Falling Edge	10	75	10	65	7	50	ns

NOTES:

1. The ratio CLKO/L1CLK must be greater than 2.5/1.
2. High impedance is measured at the 30% and 70% of V_{DD} points, with the line at $V_{DD}/2$ through 10K in parallel with 130 pF.
3. Where P = 1/CLKO. Thus, for a 25-MHz CLKO rate, P = 40 ns.

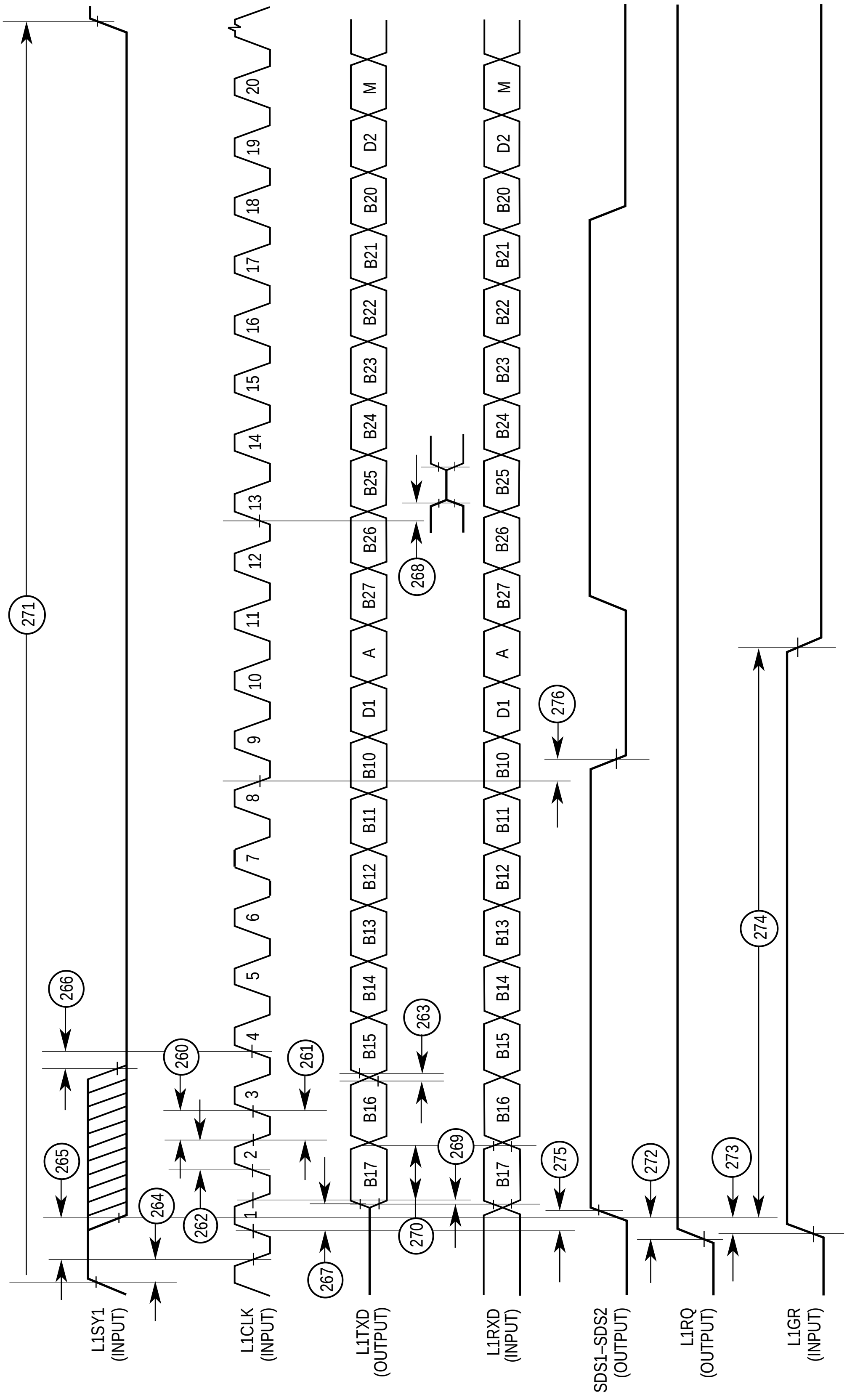


Figure 7-20. IDL Timing Diagram

7.6.16 IMP AC Electrical Specifications—GCI Timing

GCI supports the NORMAL mode and the GCI channel 0 (GCN0) in MUX mode. Normal mode uses 512 kHz clock rate (256K bit rate). MUX mode uses 256 x n – 3088 kbs (clock rate is data rate x 2). The ratio CLKO/L1CLK must be greater than 2.5/1 (see Figure 7-21).

Num.	Characteristic	16.67 MHz at 5.0 V		20 MHz at 5.0 V		25 MHz at 5.0 V		Unit
		Min	Max	Min	Max	Min	Max	
	L1CLK GCI Clock Frequency (Normal Mode) (see Note 1)	—	512	—	512	—	512	kHz
280	L1CLK Clock Period Normal Mode (see Note 1)	1800	2100	1800	2100	1800	2100	ns
281	L1CLK Width Low/High Normal Mode	840	1450	840	1450	840	1450	ns
282	L1CLK Rise/Fall Time Normal Mode (see Note 4)	—	—	—	—	—	—	ns
	L1CLK (GCI Clock) Period (MUX Mode) (see Note 1)	—	6.668	—	6.668	—	6.668	MHz
280	L1CLK Clock Period MUX Mode (see Note 1)	150	—	150	—	150	—	ns
281	L1CLK Width Low MUX Mode	55	—	55	—	55	—	ns
281A	L1CLK Width High MUX Mode (see Note 5)	P+10	—	P+10	—	P+10	—	ns
282	L1CLK Rise/Fall Time MUX Mode (see Note 4)	—	—	—	—	—	—	ns
283	L1SY1 Sync Setup Time to L1CLK Falling Edge	30	—	25	—	20	—	ns
284	L1SY1 Sync Hold Time from L1CLK Falling Edge	50	—	42	—	34	—	ns
285	L1TxD Active Delay (from L1CLK Rising Edge) (see Note 2)	0	100	0	85	0	70	ns
286	L1TxD Active Delay (from L1SY1 Rising Edge) (see Note 2)	0	100	0	85	0	70	ns
287	L1RxD Setup Time to L1CLK Rising Edge	20	—	17	—	14	—	ns
288	L1RxD Hold Time from L1CLK Rising Edge	50	—	42	—	34	—	ns
289	Time Between Successive L1SY1in Normal SCIT Mode	64 192	— —	64 192	— —	64 192	— —	L1CLK L1CLK
290	SDS1–SDS2 Active Delay from L1CLK Rising Edge (see Note 3)	10	90	10	75	7	60	ns
291	SDS1–SDS2 Active Delay from L1SY1 Rising Edge (see Note 3)	10	90	10	75	7	60	ns
292	SDS1–SDS2 Inactive Delay from L1CLK Falling Edge	10	90	10	75	7	60	ns
293	GCIDCL (GCI Data Clock) Active Delay	0	50	0	42	0	34	ns

NOTES:

1. The ratio CLKO/L1CLK must be greater than 2.5/1.
2. Condition C_L = 150 pF. L1TD becomes valid after the L1CLK rising edge or L1SY1, whichever is later.
3. SDS1–SDS2 become valid after the L1CLK rising edge or L1SY1, whichever is later.
4. Schmitt trigger used on input buffer.
5. Where P = 1/CLKO. Thus, for a 25-MHz CLKO rate, P = 40 ns.

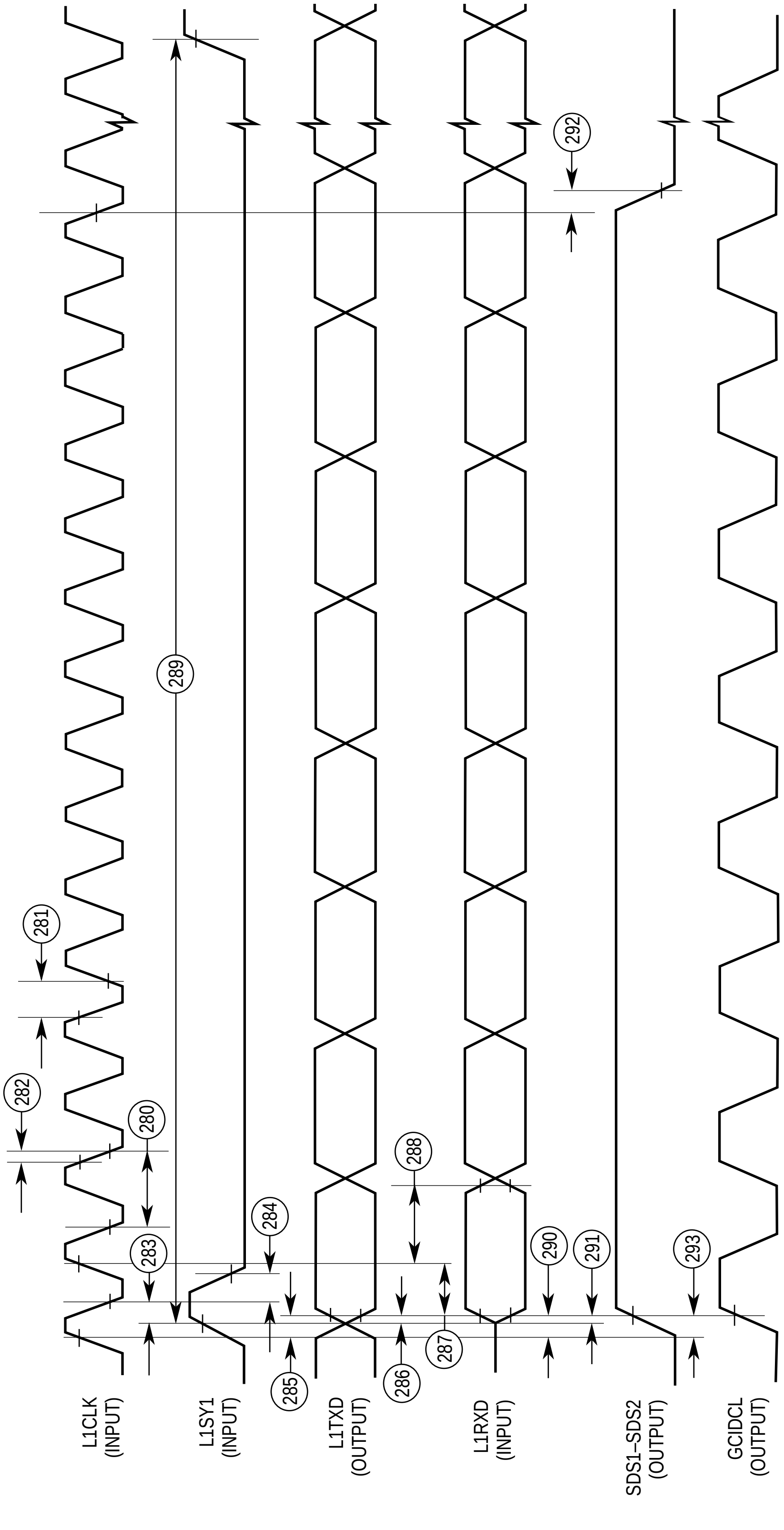


Figure 7-21. GCI Timing Diagram

7.6.17 IMP AC Electrical Specifications—PCM Timing

There are two sync types:

Short Frame—Sync signals are one clock cycle prior to the data

Long Frame—Sync signals are N-bits that envelope the data, N > 0; see Figure 7-22 and Figure 7-23).

Num.	Characteristic	16.67 MHz at 5.0 V		20 MHz at 5.0 V		25 MHz at 5.0 V		Unit
		Min	Max	Min	Max	Min	Max	
300	L1CLK (PCM Clock) Frequency (see Note 1)	—	6.66	—	8.0	—	10.0	MHz
301	L1CLK Width Low	55	—	45	—	37	—	ns
301A	L1CLK Width High (see Note 4)	P+10	—	P+10	—	P+10	—	ns
302	L1SY0–L1SY1 Setup Time to L1CLK Rising Edge	0	—	0	—	0	—	ns
303	L1SY0–L1SY1 Hold Time from L1CLK Falling Edge	40	—	33	—	27	—	ns
304	L1SY0–L1SY1 Width Low	1	—	1	—	1	—	L1CLK
305	Time Between Successive Sync Signals (Short Frame)	8	—	8	—	8	—	L1CLK
306	L1TxD Data Valid after L1CLK Rising Edge (see Note 2)	0	70	0	60	0	47	ns
307	L1TxD to High Impedance (from L1CLK Rising Edge)	0	50	0	42	0	34	ns
308	L1RxD Setup Time (to L1CLK Falling Edge) (see Note 3)	20	—	17	—	14	—	ns
309	L1RxD Hold Time (from L1CLK Falling Edge) (see Note 3)	50	—	42	—	34	—	ns

NOTES:

1. The ratio CLK/L1CLK must be greater than 2.5/1.
2. L1TxD becomes valid after the L1CLK rising edge or the sync enable, whichever is later, if long frames are used.
3. Specification valid for both sync methods.
4. Where P = 1/CLKO. Thus, for a 25-MHz CLKO rate, P =40 ns.

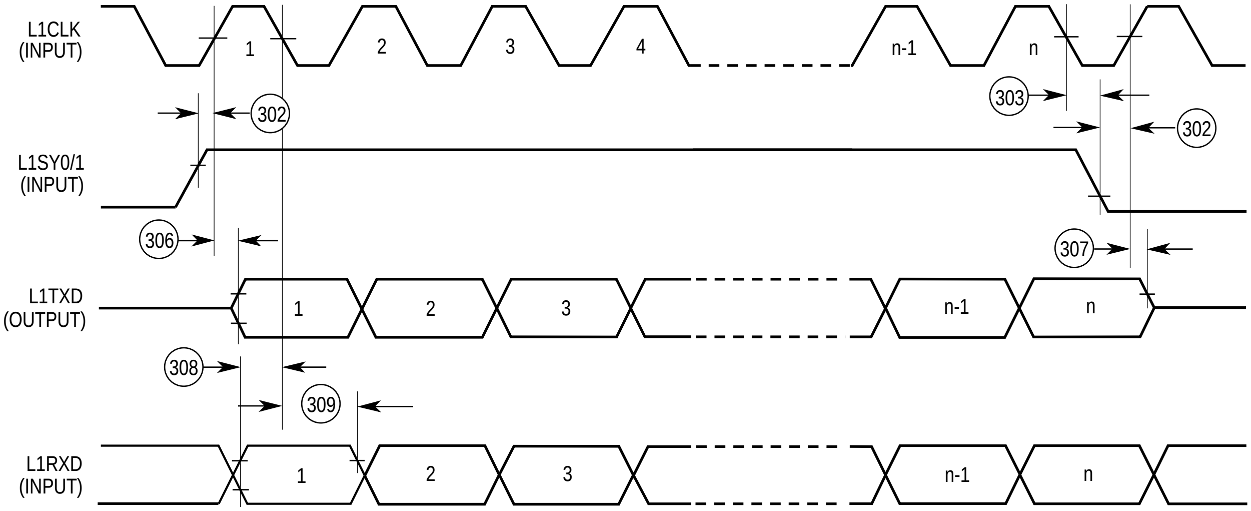
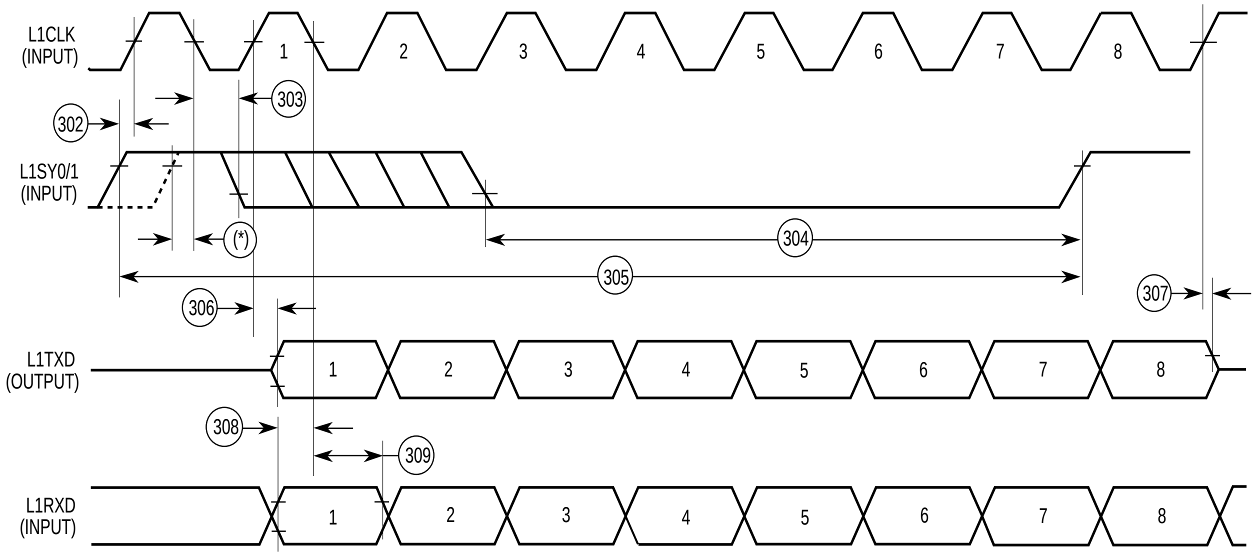


Figure 7-22. PCM Timing Diagram (SYNC Envelopes Data)



NOTE: (*) If L1SYn is guaranteed to make a smooth low to high transition (no spikes) while the clock is high, setup time can be defined as shown (min 20 ns).

Figure 7-23. PCM Timing Diagram (SYNC Prior to 8-Bit Data)

7.6.18 IMP AC Electrical Specifications—NMSI Timing

The NMSI mode uses two clocks, one for receive and one for transmit. Both clocks can be internal or external. When the clock is internal, it is generated by the internal baud rate generator and it is output on TCLK or RCLK. All the timing is related to the external clock pin. The timing is specified for NMSI1. It is also valid for NMSI2 and NMSI3 (see Figure 7-24).

Num.	Characteristic	16.67 MHz		16.67 MHz		20 MHz		20 MHz		25 MHz		25 MHz		Unit
		Internal Clock		External Clock		Internal Clock		External Clock		Internal Clock		External Clock		
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
315	RCLK1 and TCLK1 Frequency (see Note 1)	—	5.55	—	6.668	—	6.66	—	8	—	8.33	—	10	MHz
316	RCLK1 and TCLK1 Low (see Note 4)	65	—	P+10	—	55	—	P+10	—	45	—	P+10	—	ns
316a	RCLK1 and TCLK1 High	65	—	55	—	55	—	45	—	45	—	35	—	ns
317	RCLK1 and TCLK1 Rise/Fall Time (see Note 3)	—	20	—	—	—	17	—	—	—	14	—	—	ns
318	TXD1 Active Delay from TCLK1 Falling Edge	0	40	0	70	0	30	0	50	0	25	0	40	ns
319	$\overline{\text{RTS1}}$ Active/Inactive Delay from TCLK1 Falling Edge	0	40	0	100	0	30	0	80	0	25	0	65	ns
320	$\overline{\text{CTS1}}$ Setup Time to TCLK1 Rising Edge	50	—	10	—	40	—	7	—	35	—	7	—	ns
321	RXD1 Setup Time to RCLK1 Rising Edge	50	—	10	—	40	—	7	—	35	—	7	—	ns
322	RXD1 Hold Time from RCLK1 Rising Edge (see Note 2)	10	—	50	—	7	—	40	—	7	—	35	—	ns
323	$\overline{\text{CD1}}$ Setup Time to RCLK1 Rising Edge	50	—	10	—	40	—	7	—	35	—	7	—	ns

NOTES:

1. The ratio CLKO/TCLK1 and CLKO/RCLK1 must be greater than or equal to 2.5/1 for external clock. The input clock to the baud rate generator may be either an internal clock or TIN1, and may be as fast as EXTAL. However, the output of the baud rate generator must provide a CLKO/TCLK1 and CLKO/RCLK1 ratio greater than or equal to 3/1. In asynchronous mode (UART), the bit rate is 1/16 of the TCLK1/RCLK1 clock rate.
2. Also applies to $\overline{\text{CD}}$ hold time when $\overline{\text{CD}}$ is used as an external sync in BISYNC or totally transparent mode.
3. Schmitt triggers used on input buffers.
4. Where P = 1/CLKO. Thus, for a 25-MHz CLKO rate, P =40 ns.

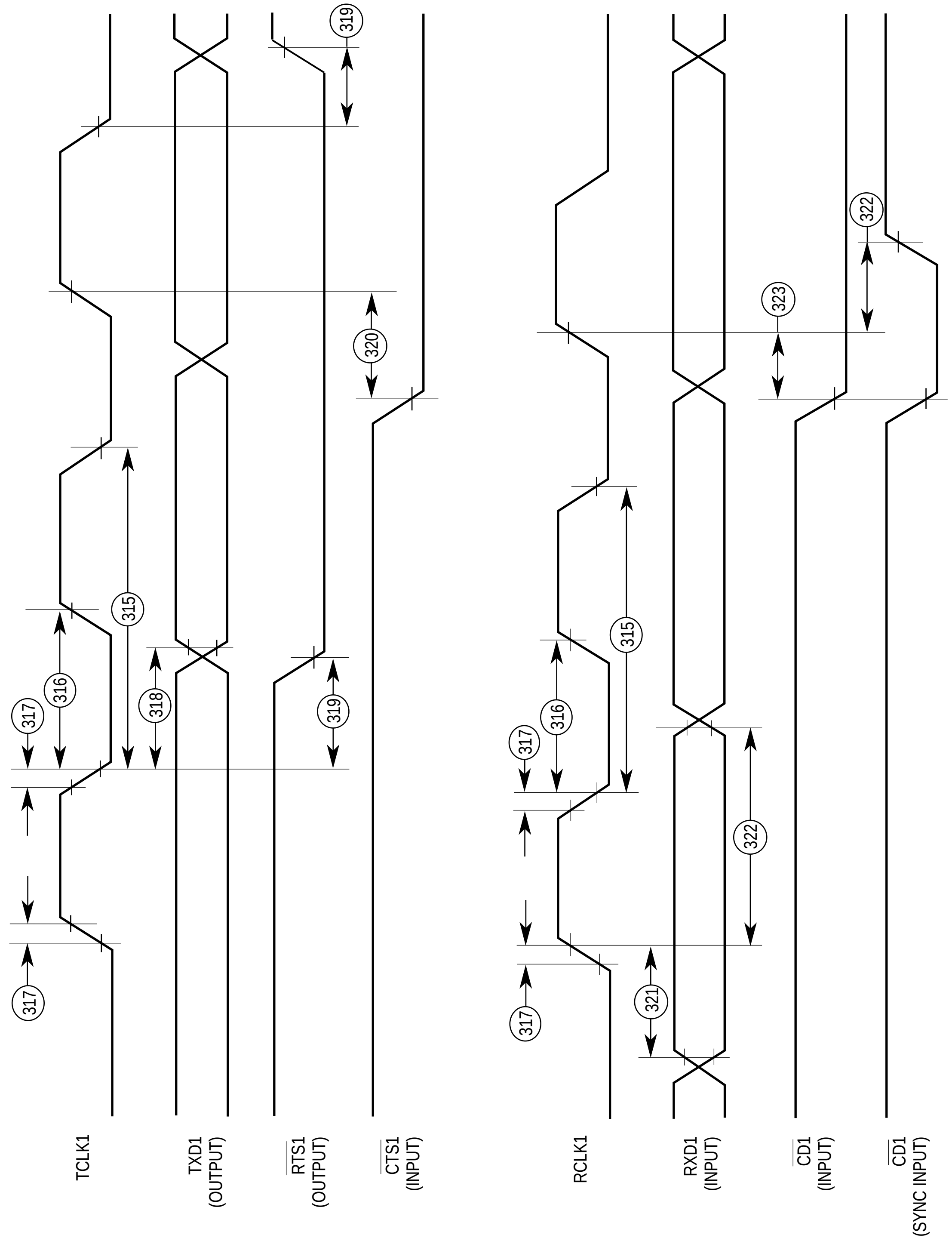

TCLK1
TXD1 (OUTPUT)
$\overline{\text{RTS1}}$ (OUTPUT)
$\overline{\text{CTS1}}$ (INPUT)
RCLK1
RXD1 (INPUT)
$\overline{\text{CD1}}$ (INPUT)
$\overline{\text{CD1}}$ (SYNC INPUT)
315
316
317
317
318
319
319
320
315
316
317
317
321
322
322
323

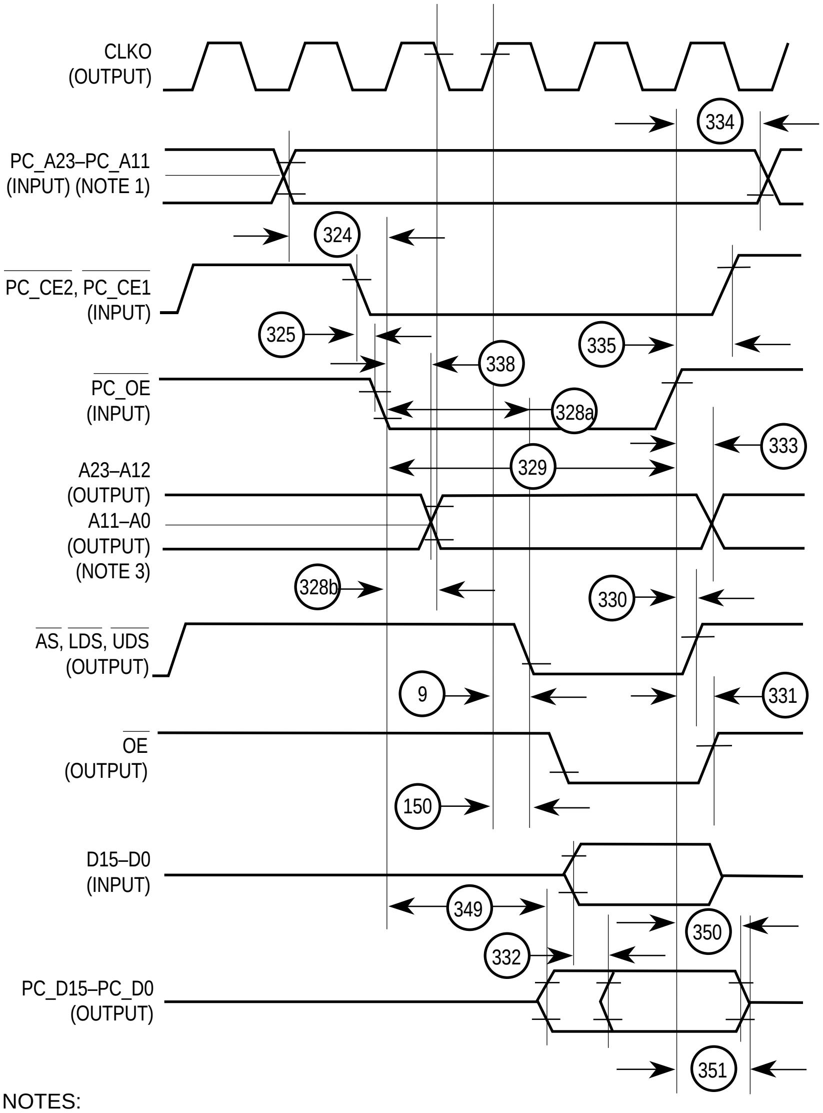
Figure 7-24. NMSI Timing Diagram

7.6.19 AC Electrical Specifications—PCMCIA Interface

Num.	Characteristic	5.0 V		Unit
		Min	Max	
324	PC Address Setup before PC_OE or PC_WE Low	10		ns
325	PC_CE Setup before PC_OE or PC_WE Low	0		ns
326	PC_CE Hold from PC_WE or PC_OE High (See Note 1)	20		ns
327	PC_Address Hold from PC_OE or PC_WE High (See Note 1)	20		ns
328a	PC_WE or PC_OE Low to AS Low (only for FAST Burst Mode)	1	2.5+ 20	clk ns
328b	Asynchronous Input Setup	10		ns
329	Minimum PC_OE or PC_WE Width (only for FAST Burst Mode) (See Note 2)	3.5		clk
330	AS, LDS, UDS Hold from PC_OE or PC_WE High (only for FAST Burst Mode)	0	20	ns
331	PC_OE High to OE High (only for FAST Burst Mode) or PC_WE High to WEH, WEL High (only for FAST Burst Mode)		25	ns
332	68k Data Bus Valid to PC_Data Bus Valid (See Note 3)		25	ns
333	PC_OE or PC_WE High to 68k Address Bus Invalid (only for FAST Burst Mode)		25	ns
334	PC_Address Hold from PC_OE or PC_WE High (See Note 4)	20		ns
335	PC_CE Hold from PC_OE or PC_WE High (See Note 4)	20		ns
336	Clock High to PC_ABUF Low		20	ns
337	Clock High to PC_ABUF High		20	ns
338	PC_OE or PC_WE Low to 68k Address Bus Valid (only for FAST Burst Mode)		30	ns
339	AS Low to 68k Data Out (only for FAST Burst Mode)	0	20	ns
340	PC_Data Valid to 68k Data Valid		25	ns
341	WEL,WEH High to 68k Data Invalid (only for FAST Burst Mode)	0		ns
342	PC_Data Setup before PC_WE High	30		ns
345	PC_Data Bus Hold Time from PC_OE High	20		ns
346	PC_OE or PC_WE or PC_IORD or PC_IOWR Low to PC_WAIT Low		30	ns
347	PC_OE or PC_WE or PC_IORD or PC_IOWR High from PC_WAIT High (See Note 5)	0		
348	Clock High to PC_WAIT High		20	ns
349	PC_OE Low to PC_Data Driven	5		ns
350	PC_OE High to PC_Data Invalid	0		ns
351	PC_Data High-Z from PC_OE High		30	ns
352	PC_Address Setup before IOWR or IORD	5		ns
353	PC_CE and PC_REG Setup before IOWR or IORD	5		ns
354	PC_OE Low to PC_Data Valid		40	ns
355	Clock Low to PC_Data Valid		25	ns
356	Minimum Pulse Width of PC_IOWR		2.5	clk
357	PC_WAIT High to PC_Data Valid (See note 6)		0	ns
358	PC_IORD High to PC_Data Invalid	0		ns
359	PC_IORD Low to PC_Data Valid		1.5+ 30	clk ns

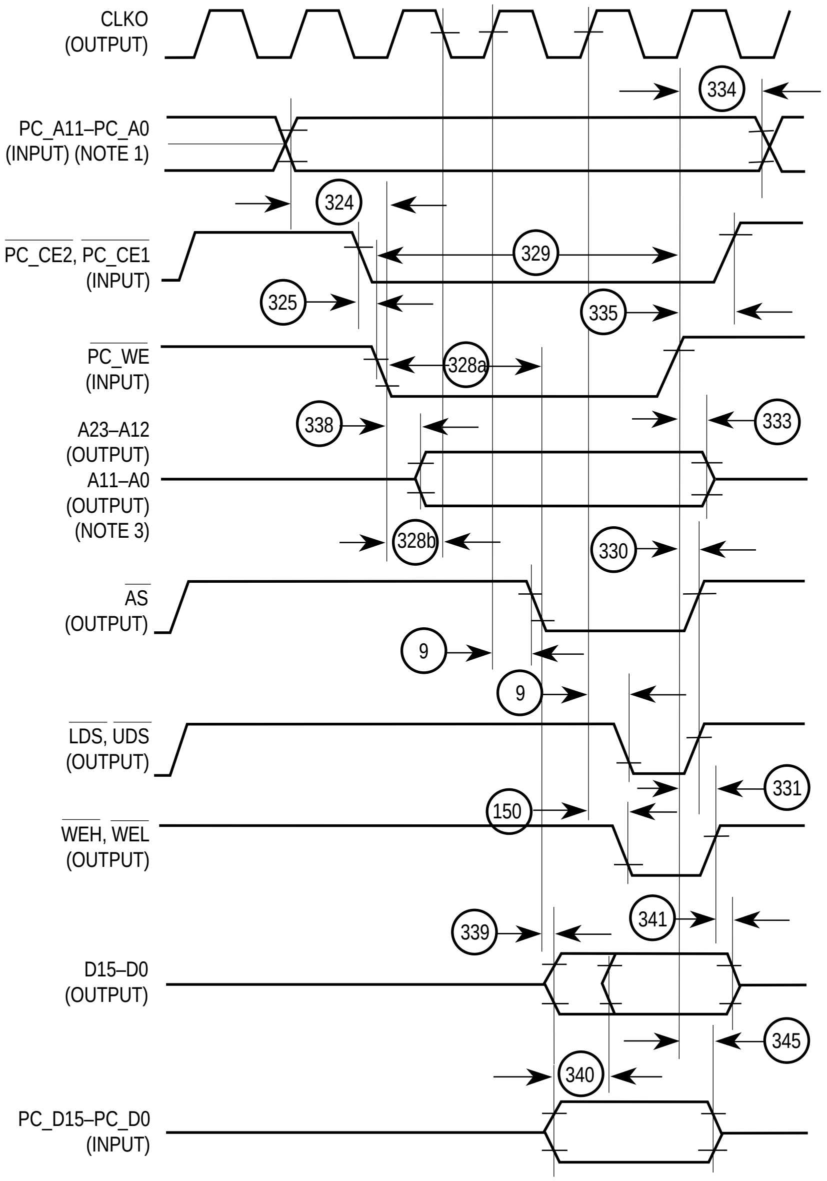
360	PC_IORD High to PC_data High-Z		30	ns
361	PC_Data Setup before PC_IOWR Low	0		ns
362	PC_Data Hold from PC_IOWR High	0		ns
363	PC_Address, PC_CE Hold from PC_IORD or PC_IOWR High	10		ns
364	PC_REG Hold from PC_IORD or PC_IOWR High	0		ns

NOTES:
1. Hold Time Required for Register Accesses Only
2. If the Asynchronous Input Setup (#328) is met this value is 1.5 clks+10ns
3. Delay through chip assuming chip is driving PC_Data Bus
4. Hold Time Required for PCMCIA to 68k Bus Accesses
5. Max is Determined by PC Host Timing
6. PC_IO Cycles with PC_WAIT Enabled



NOTES:
1. PC_A11 is input if ABUF/PC_A11 bit in PCMR = 0
2. Asynchronous input; if set-up is met AS is asserted as shown
3. If ABUF/A11 = 1 in PCMR A23–A12 are input to chip

Figure 7-25. PCMCIA to 68K Fast Burst Read



NOTES:
1. PC_A11 is input if ABUF/PC_A11 bit in PCMR = 0
2. Asynchronous input; if set-up is met AS is asserted as shown
3. If ABUF/A11 = 1 in PCMR A23–A12 are input to chip

Figure 7-26. PCMCIA to 68K Fast Burst Write

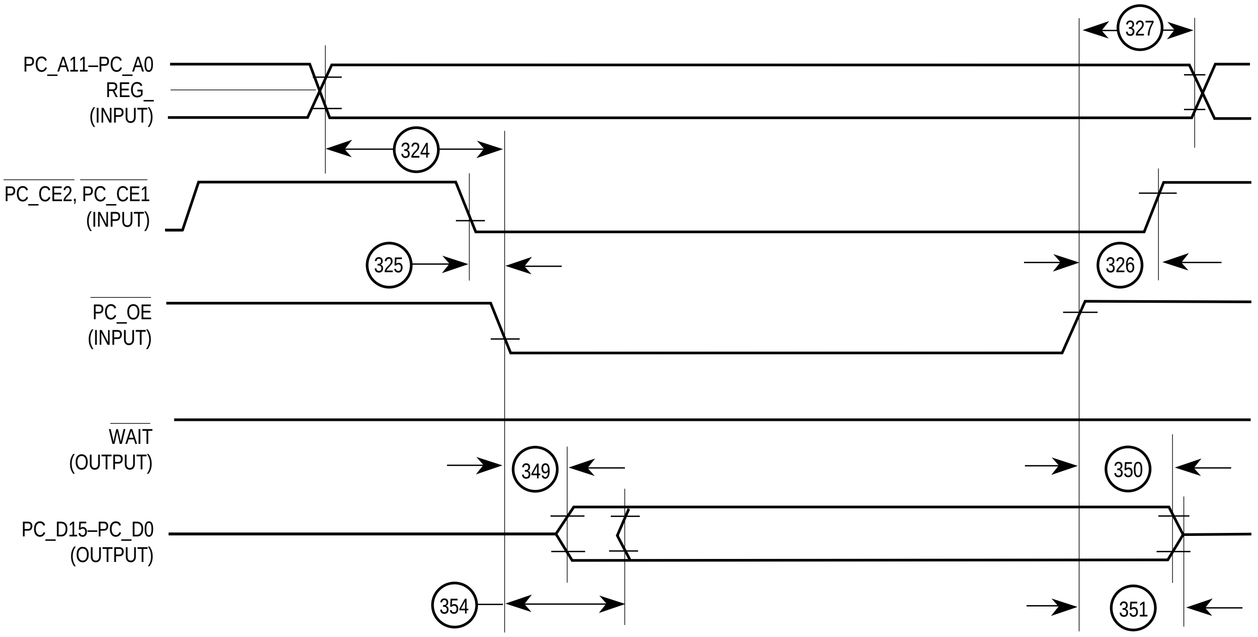


Figure 7-27. PCMCIA Register Read

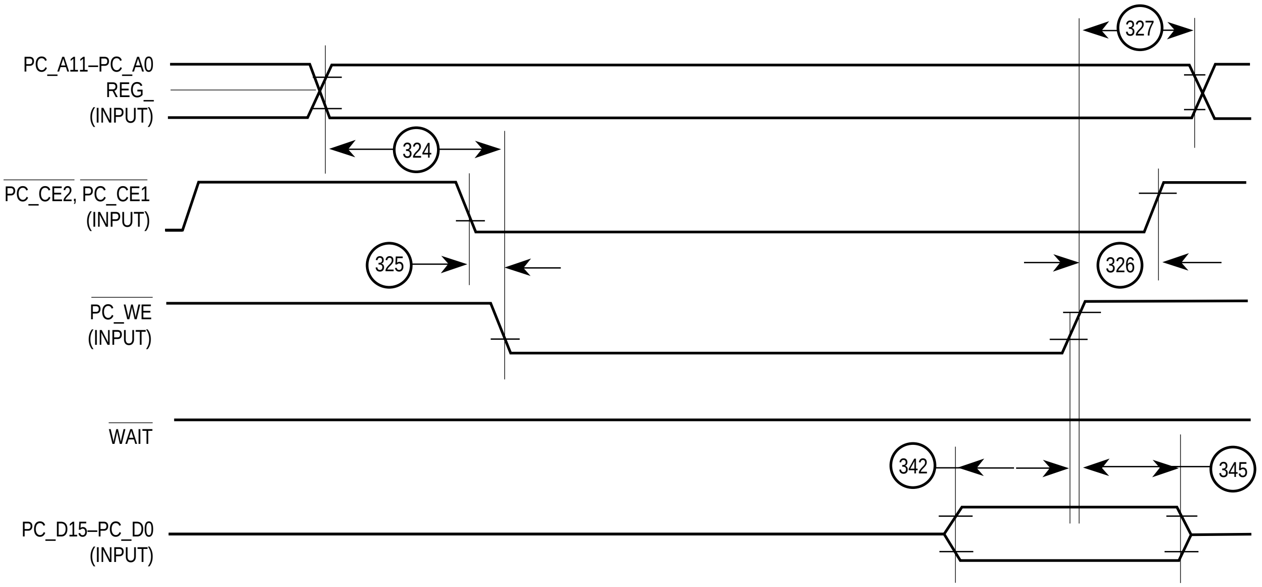


Figure 7-28. PCMCIA Register Write

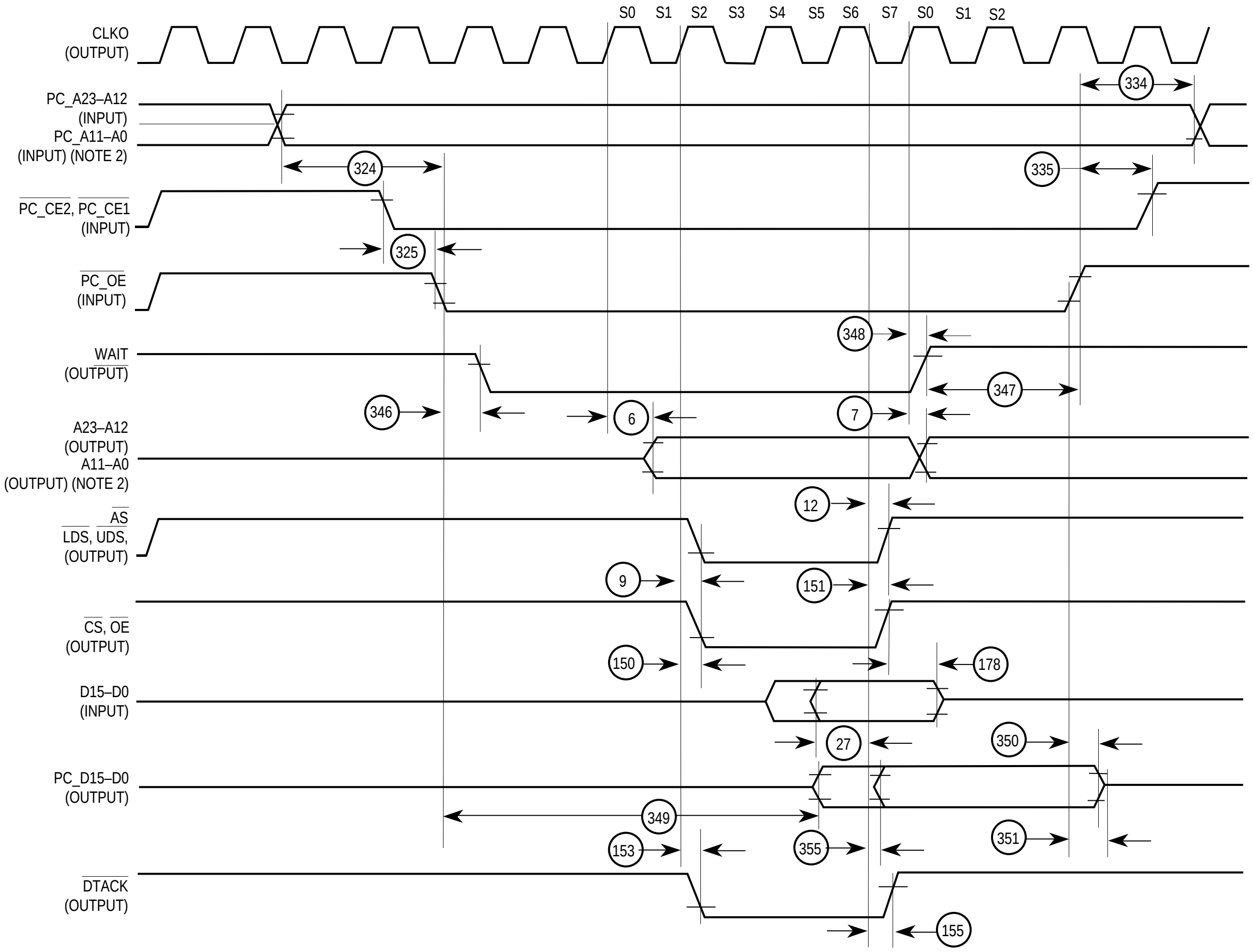
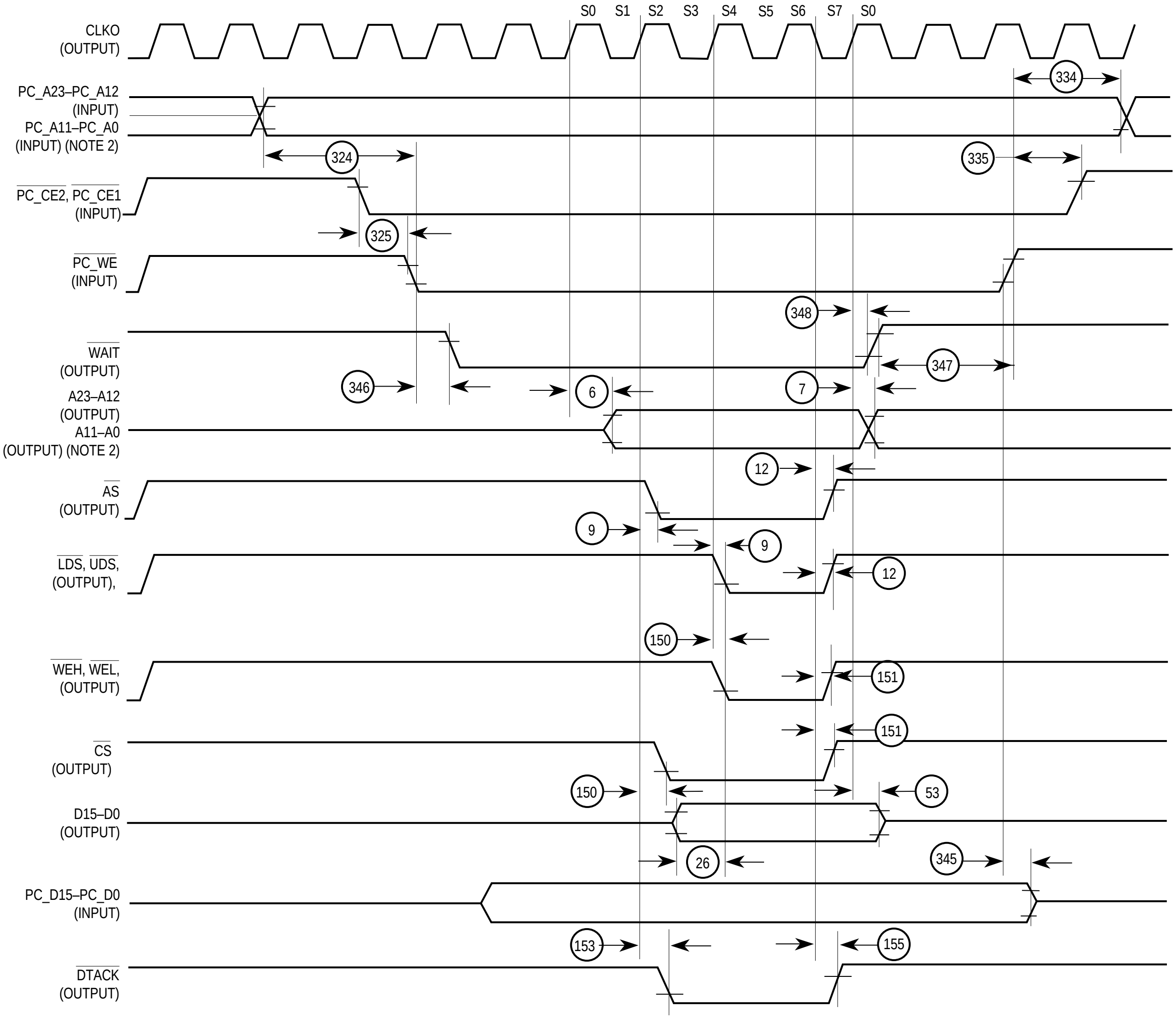


Figure 7-29. PCMCIA Normal Read



NOTES:
1. PC_A11 is input if ABUF/PC_A11 bit in PCMR=0
2. If ABUF/A11=1 in pcmr A23–A12 are input to chip
3. Asserted if ABUF/A11 = 1 in PCMR

Figure 7-30. PCMCIA Normal Write

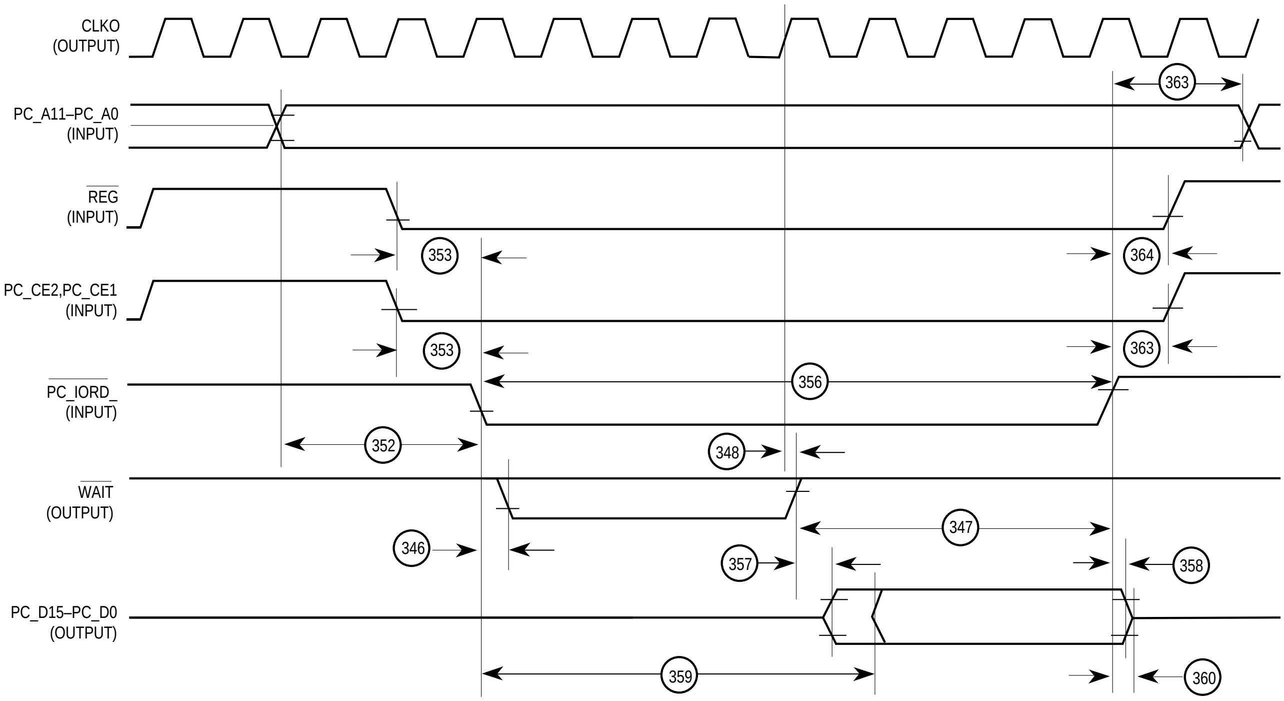


Figure 7-31. PCMCIA I/O (16550 Emulation) Read

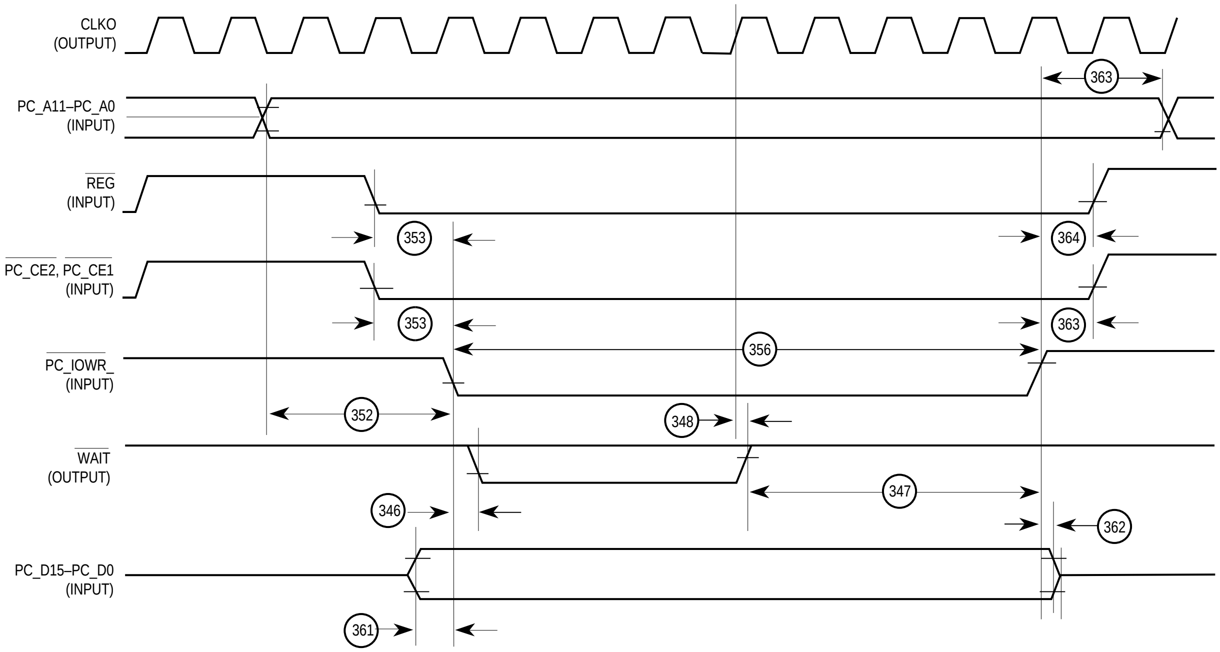


Figure 7-32. PCMCIA I/O (16550 Emulation) Write

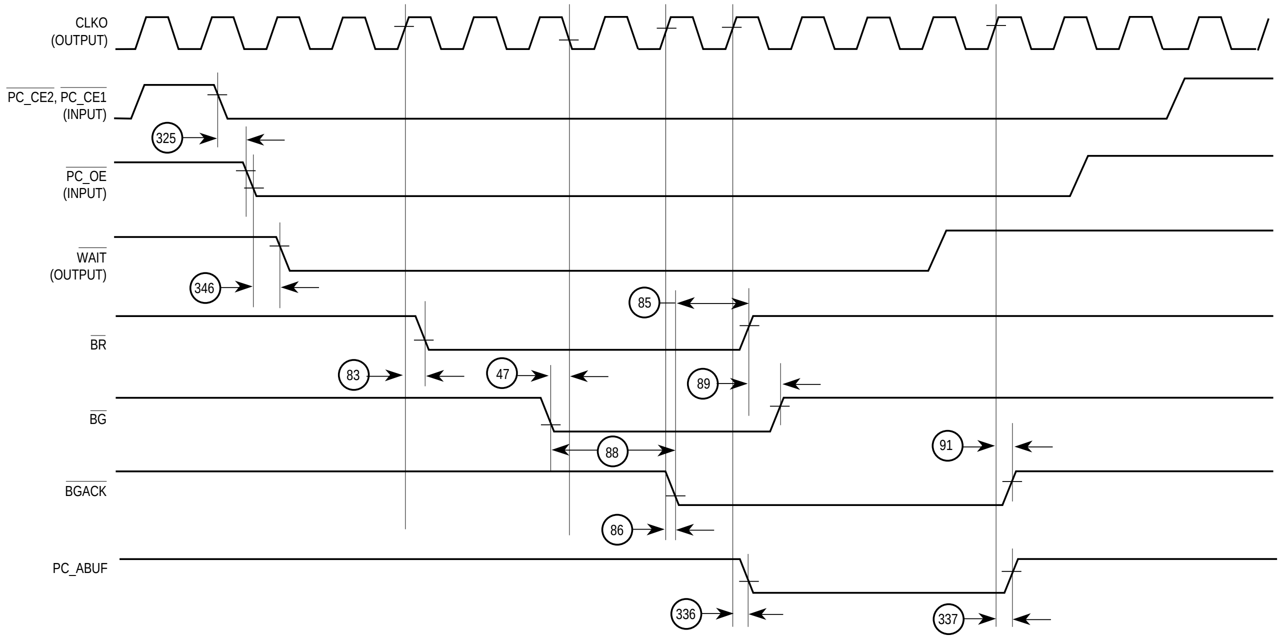


Figure 7-33. PCMCIA to 68K Arbitration

INDEX

Home Page:
www.freescale.com
email:
support@freescale.com
USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
(800) 521-6274
480-768-2130
support@freescale.com
Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com
Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku
Tokyo 153-0064, Japan
0120 191014
+81 2666 8080
support.japan@freescale.com
Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate,
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com
For Literature Requests Only:
Freescale Semiconductor
Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
(800) 441-2447
303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor
@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.
Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters which may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals” must be validated for each customer application by customer’s technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.